稀疏前沿: Transformer 大型语言模型中的稀疏注意力权衡

Piotr Nawrot*
University of Edinburgh

Robert Li
Cohere

Renjie Huang
Cohere

Sebastian Ruder†
Meta

Kelly Marchisio
Cohere

Edoardo M. Ponti
University of Edinburgh

Abstract

稀疏注意力为在 Transformer 大型语言模型中扩展长上下文能力提供了一种有前景的策略，但其可行性、效率-准确性权衡以及系统化的扩展研究尚未被探索。为了解决这一差距，我们在各种模型规模、序列长度和稀疏度水平上对多样化的长序列任务集合——包括依赖自然语言但仍可控且易于评估的新任务——进行训练无关稀疏注意力方法的仔细比较。根据我们的实验，我们报告了一系列关键发现：1）等 FLOPS 分析显示，对于非常长的序列，更大且高度稀疏的模型优于较小且密集的模型。2）在解码期间能够在统计上保证准确性保持的稀疏度水平高于预填充阶段，并在前者中与模型大小相关。3）没有明确的策略能够在所有任务和阶段中表现最佳，不同的场景需要不同的稀疏化单位或预算自适应性。即使是中等稀疏度水平，通常也会导致至少一个任务的显著性能下降，突显出稀疏注意力不是一个普遍的解决方案。4）我们引入并验证了专门针对稀疏注意力的新扩展定律，提供了证据表明我们的发现可能在我们的实验范围之外也成立。通过这些见解，我们展示了稀疏注意力是增强 Transformer 大型语言模型处理更长序列能力的关键工具，但需要对性能敏感应用的权衡进行仔细评估。

1 介绍

在大型语言模型（LLMs）中建模长序列的能力是长上下文处理和推理时缩放的核心。实现这一能力的基本瓶颈是 Transformer LLMs 中的自注意力机制：在推理时预填充过程中，注意力的复杂度随着序列长度呈二次方增长，因此导致获取第一个令牌的时间和部署成本骤增。在推理时解码过程中，密集的注意力导致键值（KV）项的缓存线性增长，其大小与序列长度成正比。因此，运行时主要受限于从内存中加载这些 KV 对的高带宽内存访问。

稀疏注意机制旨在通过使用一部分键-查询交互来近似密集注意输出，从而解决上述挑战并减少计算开销 (Fu, 2024) 。尽管此类方法有望在加速长序列建模和减少内存负载的同时保持密集模型性能 (Jiang et al., 2024) ，但由于缺乏对最先进方法的全面大规模评估，其可行性和稳健性尚不明确。尽管 Li et al. (2025b) 、Liu et al. (2025b) 和 Yuan et al. (2024) 对此问题进行了初步探索，但它们仅限于狭窄的配置范围（模型大小、序列长度和稀疏性水平）、特定的使用场景（如多轮解码），并且依赖于序列长度可变的数据集，这妨碍了序列长度依赖效应的系统分析。

* Research conducted during an internship at Cohere. Correspondence email: piotr.nawrot@ed.ac.uk
† Work done prior to joining Meta.

Preprint. Under review.

在这项工作中，我们进行了迄今为止最大规模的无训练[2]稀疏注意力方法的实证分析，涉及参数在 7B 到 72B 之间的模型，长度在 16K 到 128K 标记之间的序列，以及稀疏度在 0 % 和 95 % 之间的情况。为了实现对比分析的控制，我们首先调查了现有方法，解决比较快速发展的方法时其实现细节常常掩盖其核心设计原则的挑战。我们将这些方法提炼为四个关键方向：稀疏化的单位（块/页面或垂直和斜杠）、重要性估算（固定或上下文相关）、跨层的预算分配（统一或自适应）和 KV 缓存管理（驱逐或完整缓存）。基于这一分类法，我们选择了跨越这些设计维度的六种代表性模式，统一它们的实现，允许我们严格评估每一项基本原则的不同效果。

为了我们的评估，我们整理了一个由 9 个长上下文任务组成的基准套件，旨在系统地探讨关键因素对稀疏注意力性能的影响。这些因素包括多样化的任务类型（从检索到多跳跟踪和信息聚合），序列的自然性（合成或自然语言）变化，以及精确控制的序列长度。先前的工作强调了这些维度的重要性，表明它们对稀疏注意力的有效性有显著影响 (Chen et al., 2024; Liu et al., 2024) 。除了已建立的基准 (Rajpurkar et al., 2018; Pang et al., 2022; Tseng et al., 2016) ，我们引入了基于自然语言故事模板的新颖且更具挑战性的任务。这些补充了如 RULER (Hsieh et al., 2024) 等合成基准，后者测量核心技能但由于其人工性质可能无法推断自然数据上的性能。我们的任务通过在一个更为现实但完全可控的自然语言环境中实例化这些核心技能，解决了这一差距。所有这些让我们有机会解决当前仍未解决的基础问题：

RQ1: 在给定固定的计算预算的情况下，是选择一个小而密集的模型，还是一个大的具有稀疏注意力的模型？我们进行了 isoFLOPS 分析，发现答案取决于序列长度。对于短序列，无论是密度还是规模的任何增加都会提升性能；对于长序列，只有高度稀疏的配置才位于精度—FLOPS 的帕累托前沿上。

RQ2：在统计上保证致密注意力性能完全保留的情况下，我们可以将稀疏性提高到什么程度？为了评估这一点，我们开发了一个统计测试框架，类似于无分布风险控制 (Angelopoulos et al., 2022) 。我们发现，通常情况下，在解码过程中和对于更大规模的模型，可以实现更高的稀疏水平。然而，这种总体视角掩盖了一个关键警告：即使是中等程度的稀疏性，也经常导致大多数配置中至少一个任务的性能显著下降。

RQ3: 是否存在一种通用方法可以在多样化的长序列任务中始终表现更好？我们针对每个推理阶段（预填充和解码）和不同类别的任务分别解决这个问题。我们发现，没有一种稀疏注意力方法能够在各种不同的长序列任务中均表现优异，理想的稀疏化单位以及预算是否应为自适应的选择取决于具体的任务和阶段。

RQ4：我们能否建立稀疏注意力的缩放法则 (Brown et al., 2020; Kaplan et al., 2020; Hoffmann et al., 2022) ，这些法则可以推广到我们考虑范围之外的配置？当排除模型大小、序列长度或稀疏程度时，我们可以通过一个简单的对数线性模型可靠地预测它们的性能，这帮助验证了我们结果的普遍性。

总体而言，我们的研究结果支持采用稀疏注意力来增强大型语言模型处理更长序列的能力，同时强调这需要对权衡进行仔细评估，特别是对于对性能敏感的应用程序。我们在 `https://github.com/PiotrNawrot/sparse-frontier` 发布了我们的代码。

2 免训练稀疏注意力

给定一个输入序列的标记 $X \in \mathbb{R}^{n \times d_{\text{model}}}$ ，序列长度为 n ，嵌入维度为 d_{model} ，注意力机制首先将输入投影到查询、键和值表示：$Q = XW^Q, K = XW^K, V = XW^V$ ，其中 $W^Q, W^K, W^V \in \mathbb{R}^{d_{\text{model}} \times d_{\text{head}}}$ 是投影矩阵，d_{head} 是注意头的维度。为提高可读性，我们省略了多个注意头。

注意力将第 i 个标记的输出计算为每个值的加权和，其中权重表示通过注意力矩阵 $A \in \mathbb{R}^{n \times n}$ 定义的查询-键（QK）交互。

$$A_i = \text{softmax}\left(\frac{Q_i K^\top}{\sqrt{d_{\text{head}}}}\right) \tag{1}$$

每个位置的输出计算为 $O_i = \sum_{j=1}^{n} A_{ij} V_j$ ，结果为 $O \in \mathbb{R}^{n \times d_{\text{head}}}$ 。

[2]我们刻意关注这些方法，因为基于训练的方法要么需要从头开始训练 (Yuan et al., 2025) ，要么需要微调 (Nawrot et al., 2024) ——这两者在计算资源方面都是昂贵的，并且需要访问通常保密的训练数据混合。

基于 Transformer 的文本生成模型在两个不同阶段进行操作，每个阶段具有不同的计算特性，会影响注意力机制：

预填充过程一次性处理整个输入提示，并并行计算所有标记的隐藏表示。这会由于计算 Equation (1) 中完整注意力矩阵 A 而导致相对于序列长度的平方计算复杂度。

解码是按顺序生成输出标记，一次生成一个标记。由于每个生成步骤中只有一个查询，所以注意力的复杂度在序列长度上是线性的，但运行时间主要受限于从内存加载键值（KV）对的高带宽内存访问。

通过使 A 稀疏，仅计算查询-键标记交互的一个子集，稀疏注意力方法可以缓解这些计算瓶颈：减少预填充期间的计算开销和解码期间的内存传输需求。

稀疏注意力方法在计算减少和潜在内存节约方面的有效性通过两个关键指标进行量化：稀疏度，定义为未计算的注意力单元（例如，查询--键交互）数量与给定配置中密集注意力的总单元数之比，以及压缩比，定义为 $\frac{1}{1-\text{sparsity}}$ 。例如，稀疏度 $= 0$ 对应于密集注意力，其中计算了所有查询--键交互，而稀疏度 $= 0.9$ 表示跳过了 90% 的交互，从而导致 $10\times$ 的压缩比。

我们将现有的无训练稀疏注意力方法的广泛种类简化为四个主要维度：稀疏化单元 、重要性估计 、预算分配 和 KV 缓存管理 。我们排除令牌合并方法 (Wang et al., 2024; Nawrot et al., 2024) ，因为它们利用的是令牌相似性而非稀疏性。示例模式的可视化展示在 **??** 中。

2.1 稀疏化单元

稀疏注意力方法主要区别在于它们剪枝或保留的注意力矩阵结构单元。常见的单元包括局部窗口（每个查询周围的连续区域）、纵向列（全局可用于所有查询的 tokens）、斜线（距每个查询固定偏移的 tokens），以及块（注意力矩阵的固定大小的区块，例如 64×64 的 tokens）。较大的结构化单元如块或窗口通过更好的内存局部性提供更高的计算效率，而较小的单元则允许对重要信息进行更细粒度、更精确的选择。

基于块的方法选择单位块来近似全注意力。在预填充阶段，星注意力使用局部块和第一个前缀块近似注意力。MInference 的稀疏块模式 (Jiang et al., 2024) 另外为每个查询令牌块引入一组动态选择的块。在解码阶段，Quest (Tang et al., 2024) 和 InfLLM (Xiao et al., 2024a) 将 KV 缓存分成连续的页面，并为每个解码的令牌选择其中的一个子集。

垂直斜杠模式代表了另一种重要的单元类别。早期的稀疏注意力方法，如 LM-Infinite (Han et al., 2024) 和 StreamingLLM (Xiao et al., 2024b) ，利用了局部滑动窗口，并辅以全局共享的前缀标记，也被称为注意力汇聚点。在拓展这种方法方面，Tri-shape (Li et al., 2025b) 为后缀标记添加了全注意力，而 SnapKV (Li et al., 2024b) 引入了动态选择的垂直列。MInference (Jiang et al., 2024) 在此基础上，引入了局部窗口之外任意偏移的对角斜杠。[3]

2.2 重要性估计

为了确定保留哪些特定单元，可以使用固定模式——在所有输入中统一应用——或动态模式以适应正在处理的内容。固定模式不引入计算开销，但不能适应不同输入需求，而动态模式可以更好地保留模型质量，但需要额外的计算来识别重要的连接。

通过离线校准确定固定模式，以在所有输入上良好运行。StreamingLLM (Xiao et al., 2024b) 、LM-Infinite (Han et al., 2024) 和 MoA (Fu et al., 2024) 确定初始标记（注意力下沉点）的数量和局部滑动窗口的宽度。

内容感知方法通常估计 QK 单元（标记、块或对角线）的重要性，以仅保留最相关的前 k 个，从而最大化注意力分数召回率。它们使用轻量级的启发式方法，例如从最大幅度维度 (**?**) 近似的注意力分数 [SparQ, Quest;][] Ribar2023SparQAB，Tang2024QuestQS 或者块级池化的标记表示 (Jiang et al., 2024) 。一些方法对查询进行子采样 (SampleAttention; **?**) ，认识到最近的查询标记通常能更好地指示 KV 单元的重要性，如 MInference 的 Vertical-Slash 模式 (Jiang et al., 2024) 和 SnapKV (Li et al., 2024b) 。在解码过程中，来自运行平均值 (H2O; **?**) 或最新查询 (TOVA; **?**) 的聚合注意力分数引导 KV 单元的选择或驱逐，再次优先考虑可能获得最高注意力权重的单元。一些方法结合注意力得分和补充启发式方法，如键 (Devoto et al., 2024)

[3]有趣的是，为了高效地计算沿这些对角线的注意力，MInference 使用与这些对角线对齐的 64Œ64 块，而不是计算单个查询-键值对的注意力。

或值 (VATP; **?**) 的向量范数，认识到一个标记的贡献取决于其注意力得分和值向量大小，这在方程 1 中显而易见。

2.3 预算分配

稀疏注意力设计中的第三个关键维度涉及在给定目标稀疏水平的情况下如何在模型的不同组件（如层和头部）之间分配计算预算，这涉及到统一的简单性和自适应表达能力之间的基本权衡。

均匀分配假设每个头部关注相同数量的 tokens 或块，就如同 Block-Sparse 注意 (Jiang et al., 2024) 和 SnapKV (Li et al., 2024b) 。虽然这种方法在计算上更简单，但它忽略了并非所有层和头对模型准确性的贡献是相同的，其注意力分布往往表现出多样的稀疏特性 (Zhang et al., 2024) 。

诸如 PyramidKV (Cai et al., 2024) 和 PyramidInfer (Yang et al., 2024b) 等自适应方法观察到注意力得分的熵随着层深度的增加而减少，这表明早期层需要比较深层更大的预算。同样，稀疏注意力混合 (MoA; **?**) 使用一阶泰勒展开的自动预算调整程序来优化跨层分配全局注意力预算。补充这种层间的方法，Ada-KV (Feng et al., 2024) 通过选择每层的前 $(k \times h)$ 个 token（其中 h 是头的数量）来关注每一层的灵活分配，重新分配总缓存大小，以便更关键的头保留额外的键，而不太重要的头进行激进的裁剪。

基于自适应阈值的分配代表了最灵活的方法，完全放宽了固定的全局预算限制。诸如 Twilight (Lin et al., 2025) 、FlexPrefill (Lai et al., 2025) 、Tactic (Zhu et al., 2025) 和 SampleAttention (Zhu et al., 2024) 等方法建立覆盖阈值（例如，捕获注意力权重总和的 95 %），并允许每个头部动态选择所需的单元数量以达到这些阈值。因此，表现出分散（高熵）注意力的头部自然会消耗更大比例的预算，而具有尖锐注意力的头部则消耗更少的 tokens。这些方法还包含一个最小预算参数，使其在面对极端稀疏的特殊情况时具有鲁棒性 (Chen et al., 2024) 。

2.4 KV 缓存管理

区分稀疏注意力方法的最终维度涉及它们对 KV 缓存管理 的处理，这在受内存限制的解码阶段尤为关键。虽然预填充主要受益于计算速度的提升，但解码性能常常受到存储和访问过去标记的 KV 缓存所需内存的限制。用于解码的稀疏注意力方法通过两种主要策略解决这一瓶颈，形成了内存占用减少和信息保真度之间的根本权衡。

一种策略涉及 KV 缓存驱逐，其中类似 H2O (Zhang et al., 2023) 或 SnapKV (Li et al., 2024b) 的方法基于估计的重要性永久性地从缓存中丢弃选定的标记或块。这直接减少了内存占用和数据传输成本，使得在固定的内存预算内可以处理更长的序列，并有可能提高硬件利用率。然而，这种方法牺牲了信息保真度，因为如果丢弃的标记在生成过程中变得重要，则无法恢复。在识别需要驱逐的标记时是有挑战的，因为重要标记的集合在生成步骤中可以显著变化，特别是在上下文变化时 (Li et al., 2025b) 。

或者，其他方法保持完整的 KV 缓存，但通过在注意力计算过程中选择性地仅从内存中加载必要的 KV 对来优化计算，Quest (Tang et al., 2024) 和 SparQ (Ribar et al., 2024) 就是这种方法的例子。虽然这些方法可能会因用于重要性估计的辅助数据结构（例如，页面摘要）引入少量的内存开销，但它们避免了逐出操作所固有的信息损失。这通常使它们能够比基于逐出的方法在更高的稀疏级别下有效运行，尽管它们没有减少存储缓存本身的峰值内存需求。

3 实验设置

我们在 Qwen 2.5 模型上进行实验（7B, 14B, 32B, 72B 参数）。选择这一模型族是为了系统地研究模型大小如何与序列长度、任务特性和稀疏模式进行交互。Qwen 2.5 独特地支持 128k 的上下文长度，并提供通过一致的方法训练的多种模型大小——这对于控制的缩放实验来说至关重要。我们只修改了注意力机制，保留了原有的架构，并使用全 bf16 精度的 vLLM 推理引擎。每种稀疏注意力模式的实现细节在 Appendix A.1 中。

3.1 稀疏注意力方法

我们评估了一系列稀疏注意力方法，我们选择这些方法作为跨越第 2 节中描述的关键维度的代表集合。我们专注于内容感知方法，因为先前的工作已证明固定模式相对于其内容感知的对应方法表现一直不佳 (Li et al., 2025b)： 完整列表展示在 Table 1 。

	Method	Unit	Budget	KV Cache Management
Prefill	Vertical-Slash (Jiang et al., 2024)	verticals and slashes	uniform	N/A
	FlexPrefill (Lai et al., 2025)	verticals and slashes	threshold-based	N/A
	Block-Sparse (Jiang et al., 2024)	blocks	uniform	N/A
Decode	SnapKV (Li et al., 2024b)	tokens	uniform	Eviction
	Ada-SnapKV (Feng et al., 2024)	tokens	adaptive	Eviction
	Quest (Tang et al., 2024)	pages	uniform	Full Cache

Table 1: 在我们的实验中，对内容感知的稀疏注意力方法进行了全面基准测试。这些方法在单元、预算分配和 KV 缓存管理方面代表了不同的策略。

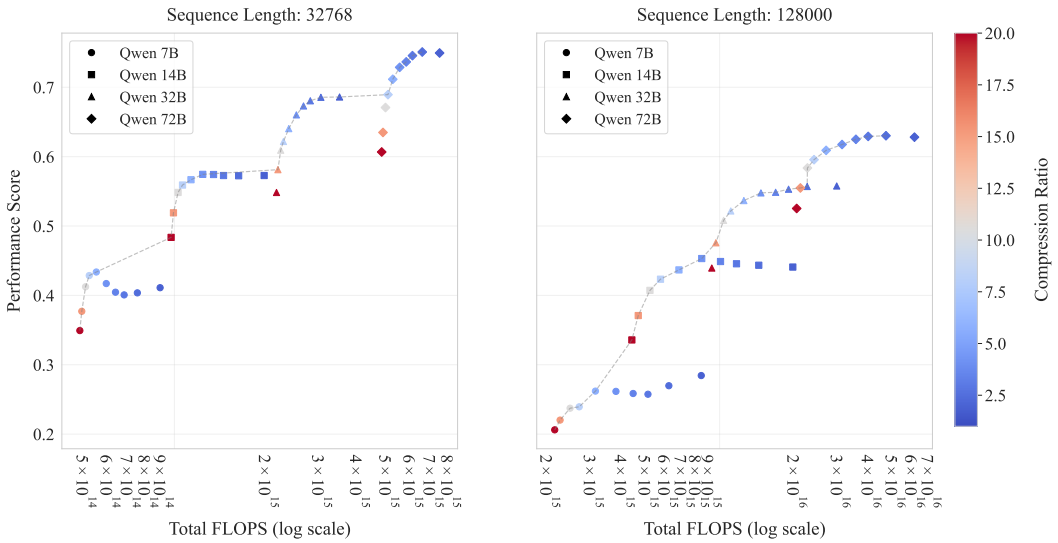
3.2 任务

我们评估了 9 项多样化任务，这些任务旨在反映沿着 3 个关键维度的不同特征，这些维度被认为会影响稀疏注意力性能：任务难度——由分散度（定位必要信息的难度）和范围（需要处理信息的量）定义 (Goldman et al., 2024) ——以及数据自然性（自然语言与合成数据）。这种多维度方法的动机来自于最近的发现，即注意力模式在不同任务类型中显著不同：检索任务通常表现出局部注意力，而推理任务则展示更均匀分布，这对稀疏方法来说具有挑战性 (Liu et al., 2025c; Chen et al., 2024; Li et al., 2025b) 。自然性维度也很重要，因为具有任意符号序列的合成任务与自然语言相比会产生不同的标记表示分布 (Liu et al., 2024) 。因此，我们的任务套件包含从 RULER 基准中选取的四项核心任务 (Hsieh et al., 2024) ——检索（NIAH）、多跳推理（VT）、聚合（CWE）和问答（SQuAD）——以提供针对特定能力的控制环境（主要是合成的）。我们补充了从现有基准中评估为污染风险最小的自然文本任务 (Li et al., 2024a) ，例如来自 QuALITY 和 TOEFL 的问答；然而，这些是低分散度、低范围的任务。因此，我们另外引入了三个新任务（故事检索、多跳、过滤）以将 RULER 的挑战性任务（具有高分散度或范围）转换为更具代表性的自然叙事，更符合现实世界的使用。我们刻意避免类似总结这样的开放性任务，因为评价指标不可靠 (Yen et al., 2024; Ye et al., 2024) ，而是专注于需要事实答案的结构化输出任务，从而通过精确匹配准确性、交集覆盖率（IoU）和 F1 评分（均为 0 到 1 范围）进行精确评估。这些任务被总结在 Table 2 中，详细描述在 Appendix A.2 中，实例展示在 Appendix G 中。

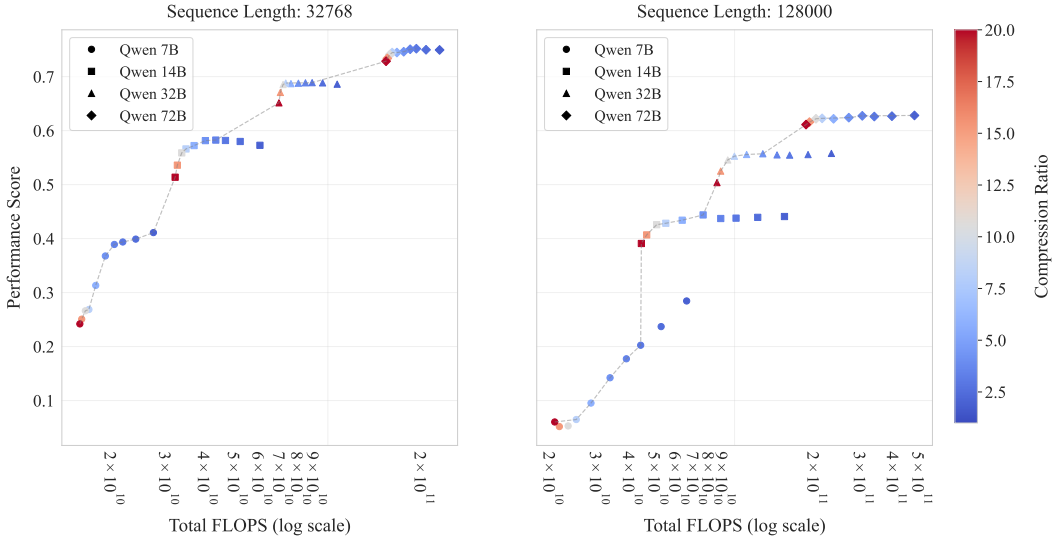
Task Name	Description	Dispersion	Scope	Natural
QA (SQuAD)	Open-ended QA on a specified document among distractors	Low	Low	✓
QA (QuALITY, TOEFL)	Multiple-choice QA on a specified document among distractors	Low	Low	✓
Ruler NIAH	Extract 4 values for specified keys among many distractor key-value pairs	Low	Low	×
Ruler VT	Identify variables that resolve to a specific value via chained assignments	High	Low	×
Ruler CWE	Identify the 10 most frequent words from a list with distractors	Low	High	×
Story Retrieval	Answer 16 factoid-style questions about specific chapters in a long narrative	Low	Low	✓
Story Multi-hop	Identify the item acquired immediately before a target item across chapters	High	Low	✓
Story Filtering	Identify chapters where no item purchases occurred in a long narrative	Low	High	✓

Table 2: 9 个评估任务的总结：QA 任务基于现有的数据集——SQuAD (Rajpurkar et al., 2018) 、QuALITY (Pang et al., 2022) 、TOEFL (Tseng et al., 2016) ——而 NIAH、VT 和 CWE 则来自 RULER 基准测试 Hsieh et al. (2024) 。剩下的三个（故事检索、多跳和过滤）是我们的贡献：我们自动生成多章节叙事来评估与 RULER 任务相同的技能，但在自然文本中表达。对于每个任务，我们指明其是否具有高度或低度分散（信息难以定位）、大或小范围（必要信息量大），以及它是基于自然文本还是合成的。

我们评估了序列长度为 16k、32k、64k 和 128k 标记的情况，每个配置使用 100 个样本。我们评估了任务、模型大小、序列长度和稀疏注意模式的所有组合，在从 1 × 到 20 × 的固定压缩比上，插值评估中间点的性能。我们确保输入样本达到目标最大标记长度的 95-100 % ，



(a) 使用垂直斜杠模式的预填充 IsoFLOPS 分析。



(b) 使用 Quest 模式的解码中 IsoFLOPS 分析。

Figure 1: 对比在批量大小为 1 的情况下的 FLOPS 性能，这是序列长度、模型大小和稀疏级别的函数。我们报告 4 种模型大小（标记）和压缩比率高达 20 × （热图）。性能得分在所有 9 项任务中进行了聚合。在图中，我们展示了两种序列长度——32k（左）和 128k（右），以及两个阶段——预填充（上）和解码（下）。重要的是，存在一个相变，即在一个关键的序列长度之后（对于 Qwen 系列模型为 32-64k 个标记），高稀疏性且大型的模型在相同的 FLOPS 预算下超过了密集且小型的模型的性能。有关我们如何估计 FLOPS 的细节，包括稀疏注意力方法的索引成本，请参见 Appendix D 。

为评估序列长度对性能的影响提供统一的基础。在 Karpinska et al. (2024) 的基础上，我们采用结构化的提示格式（见 Appendix E ），鼓励模型在给出最终答案之前通过思维链显式推理。

4 结果

4.1 等效 FLOPS 分析

研究问题 1：在给定的计算预算下，是选择一个具有较密集注意力的小模型，还是选择一个具有较稀疏注意力的大模型？

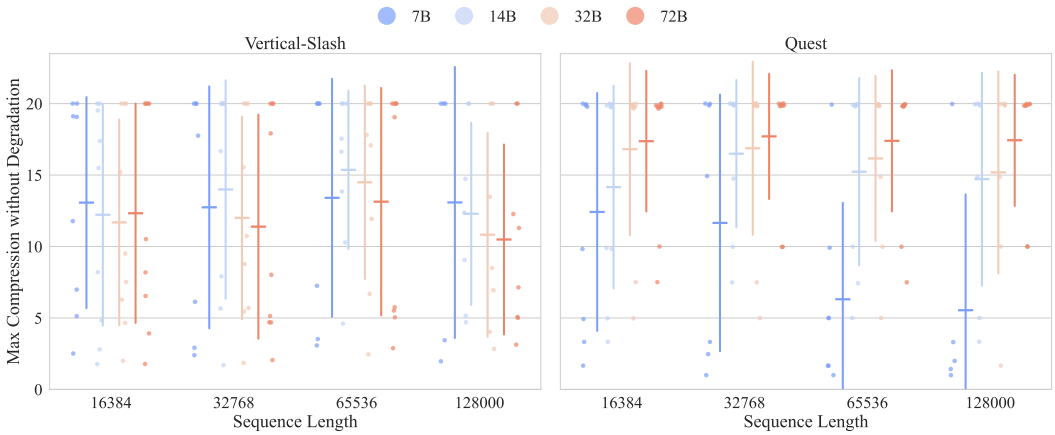


Figure 2: 在不同的模型规模（颜色）和序列长度（x 轴）下，具有统计显著性表现保持的最大压缩比（y 轴）。每个点代表一个任务，水平条显示跨任务的平均最大压缩，垂直条表示标准差。左图：预填充的垂直斜线模式。右图：解码的 Quest 模式。关键结论是解码在平均上能够忍受比预填充更高的压缩比，较大的模型即使在非常高的压缩比下也能保持性能。然而，几乎每种配置都有至少一个任务，其可容忍的最大压缩低于 5 × （唯一的例外是 72B Quest）。

Figure 1 中的结果比较了在序列长度为 32k 和 128k 标记时，不同模型大小和压缩比的任务平均性能与 FLOPS。[4] 我们观察到序列长度在准确性-FLOPS 权衡中有重要影响。

对于较短的上下文（32k tokens；1a 左图和 1b 左图），大多数模型大小--稀疏性组合位于帕累托前沿。在这种情况下，降低稀疏性能比增加模型大小更有效地带来更好的性能。这是由于在较短序列中，非注意力组件（嵌入、MLP 和输出层）主导了计算成本（参见 Appendix D 中的解释）。尽管测试压缩率高达 20 × ，我们观察到每个模型大小的 FLOPS 范围在这个序列长度下并没有交叉；即使稀疏水平不同，模型大小增加一倍（例如，14B 对比 7B）的计算成本总是超过较小模型中 MLP 和稠密注意力的组合成本。

在较长的上下文（128k 标记）中，计算动态发生了显著变化，注意力稀疏性的影响变得更加明显。图 1a 右和 1b 右表明，主要只有高稀疏性的模型位于帕累托前沿，揭示了一种效率交叉点：在相同的计算预算下，较大的稀疏模型超越了较小的密集模型。对于预填充，具有 5-15 × 压缩比的模型仍保持最佳状态，而那些具有 20 × 压缩比的模型则低于最佳边界。解码在面对高稀疏性时表现出更好的弹性，甚至 20 × 压缩配置也优于较小的模型。

仔细观察模型对稀疏性的特定敏感性，我们发现不同模型规模之间存在明显的缩放行为。在使用 Quest 模式进行解码时，性能对稀疏性恢复的能力与模型规模密切相关。Qwen 7B 在高压缩下表现出明显的性能下降，32k tokens 时从 40 %（0.4 降至 0.24），128k tokens 时从 79 %（0.29 降至 0.06），这是将稠密变体与 20 × 压缩变体相比较的结果。相比之下，较大的模型（32B, 72B）在压缩时维持了性能的稳定，即使在 20 × 压缩下，性能表现的差距也始终低于 0.05。预填充展示出显著不同的行为，在所有模型规模中，稠密和高稀疏模型之间的性能差距保持相对一致（0.1 到 0.15）。

4.2 具有性能保证的最大稀疏性

RQ2: 能够保证性能保留的最高稀疏度是多少?

为了找到每个任务可接受的最大稀疏性，我们在每种模型大小和序列长度的组合中，针对每个稀疏水平，对稠密模型及其对应的稀疏模型的性能进行单尾 Welch's t 检验。最大稀疏性是指在检验结果不显著的最高稀疏水平，显著性水平为 $p < 0.05$ 。[5]

[4]我们使用垂直斜杠模式进行预填充，并使用 Quest 模式进行解码来处理这个问题，因为这些模式在各自的推理阶段中平均表现最佳（见 Section 4.3 ）。

[5]我们选择单尾 Welch t 检验，因为我们不假设两个群体之间的方差相同，并且我们只关心性能是否显著下降。我们剔除了性能随机的模型-稀疏-任务组合，即低于阈值 0.05 的组合。

Figure 2 揭示了预填充和解码阶段的不同模式。对于使用 Vertical-Slash 的预填充，我们观察到模型大小、序列长度和可承受的最大压缩之间没有明显的相关性。在所有序列长度上，模型在跨任务平均时可以实现超过 10 × 的压缩，而性能没有显著下降。高标准差（约为 7）表明压缩容忍度在任务上存在显著的差异。

在使用 Quest 进行解码时，模型大小成为一个关键因素。事实上，7B 模型表现出较高的可变性，并且序列长度和最大压缩之间存在明显的负相关——从 16k 标记时的 12 × 开始，但在 128k 标记时下降到 5 × 。相比之下，较大的模型（32B 和 72B）保持着稳定的高压缩比（大约 17 × ），无论序列长度如何。对于这些较大的模型，大多数任务可以在不显著降级的情况下容忍测试的最大压缩比 20 × 。

一个关键的观察是，尽管这些平均值看起来很有前景，几乎每个配置都有至少一个任务，其可承受的最大压缩低于 5 × ，只有使用 Quest 的 72B 是个例外（最低 10 × ）。这突显了一个重要的局限性：尽管稀疏注意力方法平均看起来非常有效，但在每个配置中特定的输入仍可能受到稀疏化的不成比例影响。

4.3 单个任务和方法的结果

RQ3：是否存在放之四海而皆准的方法在各种长序列任务中始终表现优异?

在 Figure 3 中，我们以更细的粒度检查任务特定特征如何与预填充和解码中的各种稀疏化模式相互作用。我们观察到根据任务特征，存在明显的趋势，这在我们的实验设置中由两个轴定义：范围（需要访问多少上下文）和分散度（相关信息的分布程度）。我们首先根据这些任务特征（Section 4.3.1 and **??** ）讨论性能趋势，然后分析非均匀预算分配策略的具体影响（Section 4.3.2 ），最后总结发现以确定是否有任何方法在所有情况下都是优越的（Section 4.3.3 ）。

4.3.1 检索任务的表现（低范围，低分散）

检索任务（QuALITY、SQuAD、TOEFL、Ruler NIAH、故事检索）的特点是范围小且分散度低，通常需要从上下文中检索特定事实。对于这些任务，稀疏注意力方法显示出一个明确的模式：退化的严重程度与查询数量直接相关。实际上，单一查询的问答数据集（QuALITY、SQuAD、TOEFL）即使在 20 × 的压缩时也能保持密集水平的准确性。Ruler NIAH（4 个查询）对于预填充方法显示出轻微下降，而对于解码方法则表现出适度下降，而故事检索（16 个查询）在除 Quest 以外的所有方法中都经历了更为显著的退化。

在稀疏化的单位上进行比较，对于全局选择单个 token 的方法（Vertical-Slash、FlexPrefill、SnapKV 和 Ada-SnapKV），在有 16 个查询的故事检索任务中，性能随着压缩比线性下降——这一现象之前由 Yang et al. (2024c) 和 Chen et al. (2024) 研究过。这些方法通过对查询 tokens 子集上的注意力分数进行平均来评估关键 token 的重要性；然而，当查询增多且可检索的关键点减少时，这种选择变得更加困难。

对于稀疏化的块级单元，我们观察到预填充和解码之间的行为差异。在预填充期间，由于两个关键限制（但出于效率原因是必要的），块稀疏在多查询检索任务中表现不佳。首先，它必须一起处理查询令牌的块（可能包含多个问题），这可能导致当查询分散在块之间或多个查询集中在单个块中时的重要性估计不佳。其次，块稀疏选择的是连续的关键令牌块，而不是单个令牌，这在检索可能仅需要特定非连续令牌的信息时引入冗余。

相比之下，块级别的 Quest 模式是在解码时唯一一种在故事检索中维持接近密集性能的方法，即使在 20 × 压缩比的情况下。不同于 Block-Sparse，Quest 为每个单独的查询标记选择关键标记的块，提供了显著更高的灵活性。然而，尽管在故事检索中表现出色，Quest 在 Ruler NIAH 上的性能随压缩线性下降，表现与或劣于 Ada-SnapKV。这种退化可以归因于 Ruler NIAH 任务的合成性质，该任务主要由随机符号序列构成，缺乏底层语法或语义——这导致与自然语言 (Liu et al., 2024) 的关键表示的分布不同。这种根本性差异影响了 Ada-SnapKV 和 Quest 区分不相关标记集的能力，而 Quest 由于其更粗粒度受到进一步的不利影响。

对于需要访问更广泛上下文或整合分散在序列中的信息的任务（例如，高范围或高分散性任务，如标尺 CWE 和 VT、故事过滤和多跳任务），块级方法（如块稀疏方法和 Quest）始终与或优于全局选择方法。具体而言，块稀疏方法在标尺 VT 任务中以中等压缩率（5-10 × ）独特地保持接近基线的性能，而其他方法即使在较小压缩率（3 × ）时性能也显著下降。同样，在解码过程中，Quest 在标尺 CWE 任务中显著保持高性能。我们认为这些结果源于信息处理方式的根本差异：对于检索任务，注意力分布集中在局部窗口和普遍重要的标记上，而对于需要聚合或推理的任务，注意力分布更均匀，因此需要更灵活的选择机制。

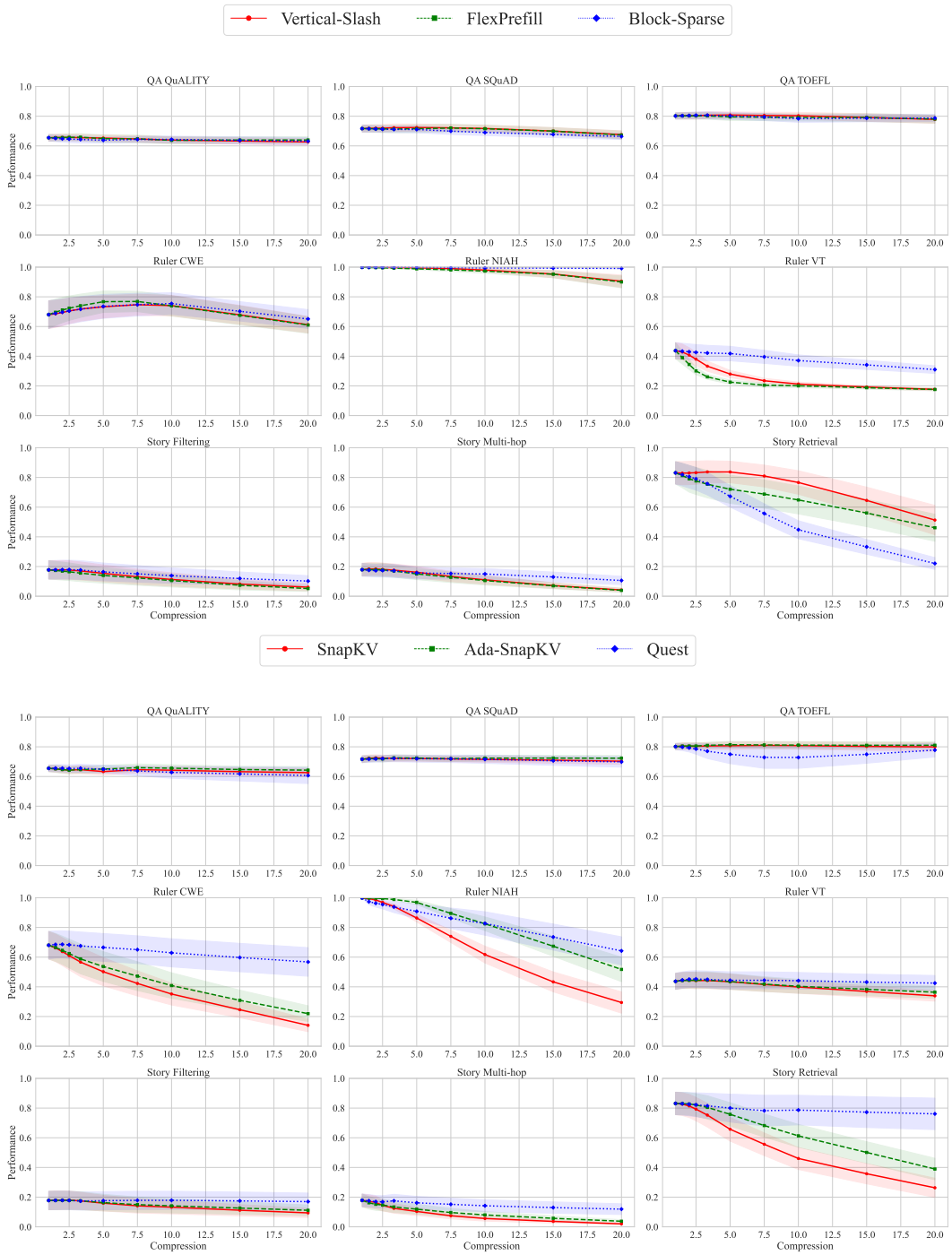


Figure 3: 不同稀疏注意力方法在 9 个任务中的性能比较，汇总在序列长度和模型上（阴影区域表示标准误差）。上图：预填充方法（Vertical-Slash, FlexPrefill, Block-Sparse）。下图：解码方法（SnapKV, Ada-SnapKV, Quest）。每个子图显示特定任务中性能与压缩之间的关系。取舍似乎极其依赖任务。总体而言，Vertical-Slash 在预填充方法中表现最佳，而 Quest 在解码方法中表现最佳。

在预填充阶段，Block-Sparse 在诸如 Ruler VT 或故事筛选等任务上表现优异，这可以通过任务结构来解释：解决独立变量链或独立检查多个章节更适合基于块的方法，相比之下，全局令牌选择方法在这些任务中难以有效分配其有限预算。对于解码，Quest 在复杂任务上表现更佳，因为它在仅优化内存传输的同时保持完整的键值缓存，不像 SnapKV 变体在生成开始前不可逆地修剪令牌。至关重要的是，这严重限制了模型通过链式推理等策略进行推理时扩展的能力，因为这些策略依赖于探索不同推理路径的能力——因此可能是不同的重要令牌集合 (Li et al., 2025b, 2024b) ——当初始方法证明不正确时 (DeepSeek-AI, 2025) 。

4.3.2 非均匀预算分配的影响

非均匀预算分配策略在推断阶段会产生不同的结果。在预填充阶段，FlexPrefill 的自适应、基于阈值的选择要么能与 Vertical-Slash 较简单的、均匀分配相匹配，要么表现不及。消融研究（Appendix A.1 ）揭示了 FlexPrefill 对最小预算超参数的极端敏感性：较低的值会降低性能，而较高的值在较高压缩率下有效地模仿了 Vertical-Slash 的固定预算方案。这种行为呼应了“注意力陷阱”现象 (Chen et al., 2024) ，在这种现象中，基于阈值的方法可能捕获了一些高注意力的标记，因此满足了它们的累计注意力召回标准，但错过了分布在注意力分布长尾的关键信息。相反，在解码过程中，Ada-SnapKV 始终优于均匀的 SnapKV，尤其是在复杂的多查询检索任务中，这些任务本质上需要更广泛的标记覆盖，展示了当生成依赖于访问多样化上下文元素时，自适应分配的好处。

4.3.3 总结与方法推荐

我们的全面评估表明，没有单一的稀疏注意力方法能在所有不同的长序列任务中表现突出，尽管提供更大灵活性的方法通常表现更好。对于预填充，Vertical-Slash 在检索任务（低范围，低分散）上始终表现最佳，但在聚合和多跳推理任务（高范围或分散）上略微落后于诸如 Block-Sparse 的块级方法。在解码方面，Quest 由于其在每个查询中选择块的高灵活性以及保持对整个 KV 缓存的访问，成为最强的方案，特别是适于处理复杂任务；然而，它在处理像 Ruler NIAH 这样的合成任务时较脆弱，并且为保持其高效选择机制所需的页级摘要而产生额外的内存开销。因此，稀疏注意力方法的最佳选择高度依赖于具体任务特征、所需的灵活性以及推理阶段（预填充与 解码）。

4.4 稀疏注意力缩放定律

RQ4: 我们能否为稀疏注意力建立缩放定律，以推广到更大的模型尺寸、序列长度或压缩比?

我们的目标是拟合刻画缩放定律，这些定律可以根据任务、模型大小、序列长度和稀疏注意压缩率预测下游准确性。我们关注每个生成阶段的表现最佳的稀疏注意模式，也就是用于稀疏预填充的 Vertical-Slash 和用于稀疏解码的 Quest。

在调查了基础模型中之前的缩放定律研究后，其中的概述可以在附录 B 中找到，我们选择了对数线性缩放定律，因为它简单地假设了一个具有任务特定截距的幂律。因此，我们的缩放定律形式为

$$s = \alpha \log N + \beta \log L + \gamma \log C + \delta_{task} + \epsilon,$$

，其中 s 是准确率的 logit（逆 sigmoid），N 是模型参数数量，L 是序列长度，C 是压缩比，δ_{task} 是任务特定截距，ϵ 是共享截距。我们在附录 C 中提供了更多关于我们缩放定律公式和拟合过程的详细信息。

我们在 Qwen 2.5 模型系列上进行我们的缩放定律的拟合和验证。我们进行了 3 次运行，每次运行中，保留一个轴上数值最高的数据点作为测试集，以评估缩放定律沿该特定轴外推的能力。具体来说，我们分别在每次运行中保留最大的模型（72 B）、最长的序列（128 k）和最高的压缩比（20× ）的数据点。

在拟合方面，我们选择 R^2 作为主要指标，因为它衡量了我们的缩放律模型解释了多少总方差。我们报告了 R^2 和在 Table 3 中推导出的参数。从 Table 3 中得出，R^2 非常高（在 0.57 到 0.74 之间）。这表明拟合非常稳健，我们选择的预测因子在很大程度上适合捕捉数据中的基本模式。此外，参数显示任务特定的交点反映了我们的任务分类：对于大多数多跳和聚合任务（Ruler VT 和 Story Filtering 和 Multi-Hop），它们的值为负。Ruler NIAH 的截距因其异常高的值而突出。

为了评估缩放规律预测的质量，对于每个外推的三个轴和九个任务中的每一个，我们在测试示例上报告两个主要指标：1）真实值与预测值之间的 Spearman 相关系数 ρ ；以及 2）预

		R^2	Parameters											
			ϵ	δ_{QA}		δ_{Ruler}			δ_{Story}			α	β	γ
				SQuAD	TOEFL	CWE	NIAH	VT	Filt.	Mult.	Retr.			
Prefill	**Sequence Length**	0.74	-16.92	0.27	0.80	0.76	14.67	-1.60	-3.55	-3.21	1.43	1.31	-1.15	-1.00
	Model Size	0.64	-18.10	0.20	0.73	-0.42	10.51	-1.63	-4.42	-3.74	0.46	1.58	-1.58	-0.83
	Compression Ratio	0.72	-18.08	0.24	0.79	0.40	13.62	-1.54	-3.65	-3.55	1.17	1.51	-1.51	-0.73
Decode	**Sequence Length**	0.64	-28.82	0.35	0.72	0.54	12.18	-0.85	-2.98	-2.87	2.32	1.72	-0.97	-0.94
	Model Size	0.57	-35.57	0.32	0.60	-0.53	8.45	-0.89	-3.88	-3.44	1.31	2.25	-1.45	-0.81
	Compression Ratio	0.64	-26.89	0.31	0.66	0.26	11.37	-0.90	-3.19	-3.40	1.78	1.81	-1.38	-0.82

Table 3: R^2 作为适应度的衡量标准和我们缩放定律的推断参数。

测准确率的平均绝对误差（MAE）。我们的缩放规律评估结果以 Spearman 相关系数 ρ 在 Table 4 中显示，并以 MAE 在 Table 7 中的 Appendix C.3 中显示。

		QA			Ruler			Story		
		Quality	SQuAD	TOEFL	CWE	NIAH	VT	Filtering	Multi-hop	Retrieval
Prefill	**Sequence Length**	.73 *	.79 *	.78 *	.86 *	.87 *	.86 *	.90 *	.34	.82 *
	Model Size	.68 *	.75 *	.60 *	.85 *	.56 *	.68 *	.89 *	.90 *	.85 *
	Compression Ratio	.77 *	.87 *	.63 *	.75 *	.82 *	.40	.79 *	.38	.65 *
Decode	**Sequence Length**	.75 *	.81 *	.69 *	.93 *	.84 *	.54 *	.67 *	.50 *	.92 *
	Model Size	.77 *	.49 *	.31 *	.92 *	.52 *	.84 *	.84 *	.89 *	.81 *
	Compression Ratio	.88 *	.91 *	.87 *	.97 *	.75 *	.92 *	.96 *	.59	.93 *

Table 4: 斯皮尔曼相关系数 ρ 按任务在每个保留的轴（序列长度、模型大小、压缩比）和每个阶段（预填充和解码）之间的真实和预测准确性之间。我们标记在 $p < 0.05$ 时具有统计显著性的条目为 * 。

基于 Table 4 ，我们发现我们的缩放定律可以有意义地预测新配置中的下游准确性，因为真实性能和预测性能之间的相关性通常很强。然而，仍然有一些设置的相关性较弱或不具有统计显著性。最值得注意的是，在故事多跳任务中，当消除压缩率（在两个阶段期间）和序列长度（在预填充期间）时，我们的定律未能外推准确性。

5 结论

我们的研究进行了迄今为止最全面的稀疏注意力方法比较，应用于一系列模型规模（最多达 72B）、序列长度（最多达 128K）和稀疏水平（最高达 95 %）跨越 9 个不同的长序列任务。我们发现，在等 FLOPS 分析的情况下，具有高稀疏性的较大模型比其较小的密集模型更有效，尤其是对于超长序列。该发现表明策略应转向结合稀疏注意力机制扩大模型规模以有效处理长序列。

另一个关键发现是，在统计上确保完全准确保留的情况下，适用的稀疏度平均而言非常高（10-15 × ），在解码过程中甚至会随着模型变大而增加。然而，这种总体视角掩盖了一个重要的警告：我们的分析结果表明，即便是中等程度的稀疏（例如，5 倍压缩）通常也会在大多数配置中导致至少一个任务上显著的性能下降。这突显出一个关键的限制：平均性能增益可能掩盖某些任务上显著的性能下降，即便在中等稀疏度下也是如此。这种对任务的敏感性强调了在各种基准上进行全面评估的必要性，涵盖潜在部署场景的全光谱。因此，稀疏注意力不是灵丹妙药，总是需要进行仔细的权衡评估，特别是对于对性能敏感的应用而言。未来的研究应优先考虑能够适应输入和任务需求的动态稀疏机制，理想情况下要结合性能保证。

深入分析结果后，我们观察到在预填充和解码阶段，通过更灵活和更细粒度地选择注意力交互，性能有提高的趋势。在预填充阶段，最佳稀疏化结构（例如块状或垂直/斜线）因任务而异，不同层之间的均匀分配表现与动态分配相当。在解码过程中，页面级的 Quest 通过保留 KV 缓存结构，避免了在生成期间因令牌修剪而导致的性能下降而表现优异。尽管更灵活和更细粒度的方法会带来较高的索引成本，但它们适应不同任务需求的能力（如预填充与解码之间及各任务之间的最佳策略不同），表明追求这种自适应稀疏机制是未来研究的一个有前景的方向。

最后，我们的研究为稀疏注意力建立了稳健的缩放法则，这些法则在未测试数据上也成立，表明观察到的趋势可能会超越测试的配置进行推广。这些见解表明稀疏注意力必将在下一代 LLM 架构中发挥关键作用，最终将有助于创造更加可扩展、高效和适应性强的 AI 模型。

6

致谢 这项工作部分得到了由 UKRI 资助的自然语言处理博士培训中心的支持，该中心由 UKRI（资助号：EP/S022481/1）以及爱丁堡大学信息学院和哲学、心理学与语言科学学院资助。

References

Samira Abnar, Harshay Shah, Dan Busbridge, Alaaeldin Mohamed Elnouby Ali, Josh Susskind, and Vimal Thilak. 2025. Parameters vs FLOPS: Scaling laws for optimal sparsity for mixture-of-experts language models. arXiv:2501.12370 .

Anastasios N. Angelopoulos, Stephen Bates, Emmanuel J. Candès, Michael I. Jordan, and Lihua Lei. 2022. Learn then test: Calibrating predictive algorithms to achieve risk control. arXiv:2110.01052 .

Dzmitry Bahdanau, Kyung Hyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In 3rd International Conference on Learning Representations, ICLR 2015 .

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. Advances in Neural Information Processing Systems .

Zefan Cai, Yichi Zhang, Bofei Gao, Yuliang Liu, Tianyu Liu, Keming Lu, Wayne Xiong, Yue Dong, Baobao Chang, Junjie Hu, and Wen Xiao. 2024. Pyramidkv: Dynamic kv cache compression based on pyramidal information funneling. arXiv:2406.02069 .

Zhuoming Chen, Ranajoy Sadhukhan, Zihao Ye, Yang Zhou, Jianyu Zhang, Niklas Nolte, Yuandong Tian, Matthijs Douze, Leon Bottou, Zhihao Jia, and Beidi Chen. 2024. Magicpig: LSH sampling for efficient LLM generation. arXiv:2410.16179 .

Leshem Choshen, Yang Zhang, and Jacob Andreas. 2024. A Hitchhiker's guide to scaling law estimation. arXiv:2410.11840 .

DeepSeek-AI. 2025. DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning. arXiv:2501.12948 .

Alessio Devoto, Yu Zhao, Simone Scardapane, and Pasquale Minervini. 2024. A simple and effective l2 norm-based strategy for kv cache compression. In Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing , pages 18476–18499.

Yuan Feng, Junlin Lv, Yukun Cao, Xike Xie, and S Kevin Zhou. 2024. Ada-kv: Optimizing kv cache eviction by adaptive budget allocation for efficient LLM inference. arXiv:2407.11550 .

Elias Frantar, Carlos Riquelme, Neil Houlsby, Dan Alistarh, and Utku Evci. 2023. Scaling laws for sparsely-connected foundation models. arXiv:2309.08520 .

Tianyu Fu, Haofeng Huang, Xuefei Ning, Genghan Zhang, Boju Chen, Tianqi Wu, Hongyi Wang, Zixiao Huang, Shiyao Li, Shengen Yan, Guohao Dai, Huazhong Yang, and Yu Wang. 2024. Moa: Mixture of sparse attention for automatic large language model compression. arXiv:2406.14909 .

Yao Fu. 2024. Challenges in deploying long-context transformers: A theoretical peak performance analysis. arXiv:2405.08944 .

Omer Goldman, Alon Jacovi, Aviv Slobodkin, Aviya Maimon, Ido Dagan, and Reut Tsarfaty. 2024. Is it really long context if all you need is retrieval? Towards genuinely difficult long context NLP. In Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing , pages 16576–16586.

Zhiyu Guo, Hidetaka Kamigaito, and Taro Watanabe. 2024. Attention score is not all you need for token importance indicator in kv cache reduction: Value also matters. In Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing , pages 21158–21166.

Chi Han, Qifan Wang, Hao Peng, Wenhan Xiong, Yu Chen, Heng Ji, and Sinong Wang. 2024. Lm-infinite: Zero-shot extreme length generalization for large language models. In Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers) , pages 3991–4008.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. 2022. Training compute-optimal large language models. In Proceedings of the 36th International Conference on Neural Information Processing Systems , pages 30016–30030.

Cheng-Ping Hsieh, Simeng Sun, Samuel Kriman, Shantanu Acharya, Dima Rekesh, Fei Jia, Yang Zhang, and Boris Ginsburg. 2024. Ruler: What's the real context size of your long-context language models? arXiv:2404.06654 .

Huiqiang Jiang, Yucheng Li, Chengruidong Zhang, Qianhui Wu, Xufang Luo, Surin Ahn, Zhenhua Han, Amir H. Abdi, Dongsheng Li, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. 2024. MInference 1.0: Accelerating pre-filling for long-context LLMs via dynamic sparse attention. In The Thirty-eighth Annual Conference on Neural Information Processing Systems .

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models. arXiv:2001.08361 .

Marzena Karpinska, Katherine Thai, Kyle Lo, Tanya Goyal, and Mohit Iyyer. 2024. One thousand and one pairs: A "novel" challenge for long-context language models. arXiv:2406.16264 .

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles .

Xunhao Lai, Jianqiao Lu, Yao Luo, Yiyuan Ma, and Xun Zhou. 2025. Flexprefill: A context-aware sparse attention mechanism for efficient long-sequence inference. In The Thirteenth International Conference on Learning Representations .

Margaret Li, Sneha Kudugunta, and Luke Zettlemoyer. 2025a. (mis)fitting: A survey of scaling laws. In Proceedings of ICLR 2025 .

Xinze Li, Yixin Cao, Yubo Ma, and Aixin Sun. 2024a. Long context vs. RAG for LLMs: An evaluation and revisits. arXiv:2501.01880 .

Yucheng Li, Huiqiang Jiang, Qianhui Wu, Xufang Luo, Surin Ahn, Chengruidong Zhang, Amir H. Abdi, Dongsheng Li, Jianfeng Gao, Yuqing Yang, and Lili Qiu. 2025b. SCBench: A kv cache-centric analysis of long-context methods. In The Thirteenth International Conference on Learning Representations .

Yuhong Li, Yingbing Huang, Bowen Yang, Bharat Venkitesh, Acyr Locatelli, Hanchen Ye, Tianle Cai, Patrick Lewis, and Deming Chen. 2024b. Snapkv: LLM knows what you are looking for before generation. arXiv:2404.14469 .

Chaofan Lin, Jiaming Tang, Shuo Yang, Hanshuo Wang, Tian Tang, Boyu Tian, Ion Stoica, Song Han, and Mingyu Gao. 2025. Twilight: Adaptive attention sparsity with hierarchical top- p pruning. arXiv:2502.02770 .

Jiaheng Liu, Dawei Zhu, Zhiqi Bai, Yancheng He, Huanxuan Liao, Haoran Que, Zekun Wang, Chenchen Zhang, Ge Zhang, Jiebin Zhang, et al. 2025a. A comprehensive survey on long context language modeling. arXiv:2503.17407 .

Xiang Liu, Zhenheng Tang, Hong Chen, Peijie Dong, Zeyu Li, Xiuze Zhou, Bo Li, Xuming Hu, and Xiaowen Chu. 2025b. Can LLMs maintain fundamental abilities under kv cache compression? arXiv:2502.01941 .

Xiang Liu, Zhenheng Tang, Hong Chen, Peijie Dong, Zeyu Li, Xiuze Zhou, Bo Li, Xuming Hu, and Xiaowen Chu. 2025c. Can LLMs maintain fundamental abilities under kv cache compression? arXiv:2502.01941 .

Xiaoran Liu, Ruixiao Li, Qipeng Guo, Zhigeng Liu, Yuerong Song, Kai Lv, Hang Yan, Linlin Li, Qun Liu, and Xipeng Qiu. 2024. Reattention: Training-free infinite context with finite attention scope. arXiv:2407.15176 .

Yuqi Luo, Chenyang Song, Xu Han, Yingfa Chen, Chaojun Xiao, Zhiyuan Liu, and Maosong Sun. 2025. Sparsing law: Towards large language models with greater activation sparsity. arXiv:2411.02335 .

Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candès, and Tatsunori Hashimoto. 2025. s1: Simple test-time scaling. arXiv:2501.19393 .

Piotr Nawrot, Adrian Łańcucki, Marcin Chochowski, David Tarjan, and Edoardo M. Ponti. 2024. Dynamic memory compression: Retrofitting LLMs for accelerated inference. In Proceedings of the 41st International Conference on Machine Learning .

Matanel Oren, Michael Hassid, Nir Yarden, Yossi Adi, and Roy Schwartz. 2024. Transformers are multi-state RNNs. In Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing , pages 18724–18741.

Richard Yuanzhe Pang, Alicia Parrish, Nitish Joshi, Nikita Nangia, Jason Phang, Angelica Chen, Vishakh Padmakumar, Johnny Ma, Jana Thompson, He He, et al. 2022. Quality: Question answering with long input texts, yes! In Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics , pages 5336–5358.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for squad. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics , pages 784–789.

Luka Ribar, Ivan Chelombiev, Luke Hudlass-Galley, Charlie Blake, Carlo Luschi, and Douglas Orr. 2024. Sparq attention: Bandwidth-efficient LLM inference. In International Conference on Machine Learning , pages 42558–42583. PMLR.

Yangjun Ruan, Chris J. Maddison, and Tatsunori Hashimoto. 2024. Observational scaling laws and the predictability of language model performance. arXiv:2405.10938 .

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. 2024. Scaling LLM test-time compute optimally can be more effective than scaling model parameters. arXiv:2408.03314 .

Jiaming Tang, Yilong Zhao, Kan Zhu, Guangxuan Xiao, Baris Kasikci, and Song Han. 2024. Quest: Query-aware sparsity for efficient long-context LLM inference. In International Conference on Machine Learning , pages 47901–47911. PMLR.

Bo-Hsiang Tseng, Sheng-Syun Shen, Hung-Yi Lee, and Lin-Shan Lee. 2016. Towards machine comprehension of spoken content: Initial toefl listening comprehension test by machine. arXiv:1608.06378 .

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. Advances in Neural Information Processing Systems , 30.

Zheng Wang, Boxiao Jin, Zhongzhi Yu, and Minjia Zhang. 2024. Model tells you where to merge: Adaptive kv cache merging for LLMs on long-context tasks. arXiv:2407.08454 .

Chaojun Xiao, Pengle Zhang, Xu Han, Guangxuan Xiao, Yankai Lin, Zhengyan Zhang, Zhiyuan Liu, and Maosong Sun. 2024a. InfLLM: Training-free long-context extrapolation for LLMs with an efficient context memory. In The Thirty-eighth Annual Conference on Neural Information Processing Systems .

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. 2024b. Efficient streaming language models with attention sinks. arXiv:2309.17453 .

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. 2024a. Qwen2.5 technical report. arXiv:2412.15115 .

Dongjie Yang, Xiaodong Han, Yan Gao, Yao Hu, Shilin Zhang, and Hai Zhao. 2024b. Pyramidinfer: Pyramid kv cache compression for high-throughput LLM inference. In Findings of the Association for Computational Linguistics ACL 2024 , pages 3258–3270.

June Yong Yang, Byeongwook Kim, Jeongin Bae, Beomseok Kwon, Gunho Park, Eunho Yang, Se Jung Kwon, and Dongsoo Lee. 2024c. No token left behind: Reliable kv cache compression via importance-aware mixed precision quantization. arXiv:2402.18096 .

Jiayi Ye, Yanbo Wang, Yue Huang, Dongping Chen, Qihui Zhang, Nuno Moniz, Tian Gao, Werner Geyer, Chao Huang, Pin-Yu Chen, Nitesh V Chawla, and Xiangliang Zhang. 2024. Justice or prejudice? Quantifying biases in LLM-as-a-judge. In Neurips Safe Generative AI Workshop 2024 .

Howard Yen, Tianyu Gao, Minmin Hou, Ke Ding, Daniel Fleischer, Peter Izsak, Moshe Wasserblat, and Danqi Chen. 2024. Helmet: How to evaluate long-context language models effectively and thoroughly. arXiv:2410.02694 .

Jiayi Yuan, Hongyi Liu, Shaochen Zhong, Yu-Neng Chuang, Songchen Li, Guanchu Wang, Duy Le, Hongye Jin, Vipin Chaudhary, Zhaozhuo Xu, Zirui Liu, and Xia Hu. 2024. Kv cache compression, but what must we give in return? A comprehensive benchmark of long context capable approaches. In The 2024 Conference on Empirical Methods in Natural Language Processing .

Jingyang Yuan, Huazuo Gao, Damai Dai, Junyu Luo, Liang Zhao, Zhengyan Zhang, Zhenda Xie, Y. X. Wei, Lean Wang, Zhiping Xiao, Yuqing Wang, Chong Ruan, Ming Zhang, Wenfeng Liang, and Wangding Zeng. 2025. Native sparse attention: Hardware-aligned and natively trainable sparse attention. arXiv:2502.11089 .

Zhenrui Yue, Honglei Zhuang, Aijun Bai, Kai Hui, Rolf Jagerman, Hansi Zeng, Zhen Qin, Dong Wang, Xuanhui Wang, and Michael Bendersky. 2025. Inference scaling for long-context retrieval augmented generation. arXiv:2410.04343 .

Yanqi Zhang, Yuwei Hu, Runyuan Zhao, John C.S. Lui, and Haibo Chen. 2024. Unifying kv cache compression for large language models with leankv. arXiv:2412.03131 .

Zhenyu Zhang, Ying Sheng, Tianyi Zhou, Tianlong Chen, Lianmin Zheng, Ruisi Cai, Zhao Song, Yuandong Tian, Christopher Ré, Clark Barrett, Zhangyang Wang, and Beidi Chen. 2023. H2o: Heavy-hitter oracle for efficient generative inference of large language models. In Proceedings of the 37th International Conference on Neural Information Processing Systems , NIPS '23, Red Hook, NY, USA. Curran Associates Inc.

Kan Zhu, Tian Tang, Qinyu Xu, Yile Gu, Zhichen Zeng, Rohan Kadekodi, Liangyu Zhao, Ang Li, Arvind Krishnamurthy, and Baris Kasikci. 2025. Tactic: Adaptive sparse attention with clustering and distribution fitting for long-context LLMs. arXiv:2502.12216 .

Qianchao Zhu, Jiangfei Duan, Chang Chen, Siran Liu, Xiuhong Li, Guanyu Feng, Xin Lv, Huanqi Cao, Xiao Chuanfu, Xingcheng Zhang, Dahua Lin, and Chao Yang. 2024. Sampleattention: Near-lossless acceleration of long context LLM inference with adaptive structured sparse attention. arXiv:2406.15486 .

A 实验细节

A.1 实现细节

本节提供了评估稀疏注意力模式的补充细节，重点是超参数调整和用于实现目标压缩比的具体配置。我们通过在 Qwen-7B 模型上进行消融研究，对每种模式的超参数进行了调整，所有任务的序列长度为 16K，压缩比从 1x 到 10x 不等。我们的主要实验在固定压缩比（1.5x, 2.0x, 2.5x, 3.33x, 5x, 7.5x, 10x, 15x, 20x）下评估了性能，并在必要时使用线性插值处理中间值。表格 6 总结了我们为每种模式、序列长度和压缩比所使用的最终参数。

A.1.1 块稀疏注意力

我们通过将注意力矩阵分成固定大小的块来实现块稀疏注意力。根据我们的消融研究（图 4），我们选择了 16x16 的块大小，因为较小的块总是能带来更好的性能。为了实现目标压缩比，我们为每个查询块选择前 k 个关键块，其中 k 是通过二分查找确定的。我们总是保留注意力汇（第一个关键块）和局部上下文（对应于查询块的对角关键块）。

A.1.2 垂直斜杠图案

我们通过为全局（垂直列）和局部（斜线对角线）注意力组件分配一个均匀预算来实现垂直斜线模式 (Jiang et al., 2024) 。我们通过使用近期查询标记的有限窗口来近似注意力分数，选择最重要的垂直线和斜线。我们的消融研究（图 10 ）揭示了任务相关的最佳近似窗口大小：检索繁重任务（Ruler NIAH, Story Retrieval）为 512 个标记，其他任务为 256 个标记。该观察结果与这些任务的典型查询长度相关（见表 5 ）。我们始终保留最初的 4 个（前缀）和最近的 64 个（局部）标记。为了实现目标压缩率，我们根据每个序列长度收集的注意力统计，计算所需的垂直线和斜线的数量。

Table 5: 各个任务在 100 个样本中问题和指令的词元长度统计，为 Vertical-Slash 和 FlexPrefill 提供近似窗口大小选择的信息。

Task	Mean Tokens	Min Tokens	Max Tokens
QA QuALITY	243.63	196	423
QA SQuAD	217.08	210	235
QA ToeflQA	237.67	202	270
RULER CWE	227.00	227	227
RULER NIAH	337.74	330	350
RULER VT	230.00	230	230
Story Filtering	184.00	184	184
Story Multi-hop	192.97	192	195
Story Retrieval	457.54	452	462

我们实现了 FlexPrefill，(Lai et al., 2025) 它通过引入每层和每个头部的动态预算分配来增强 Vertical-Slash，该分配由覆盖参数 α 和最小预算（`min_budget` ）控制。在我们的实验中，我们设定 $\tau = 0$ ，因此禁用了 Query-Aware 注意力。这个选择基于两个关键考虑因素：首先，我们的初步测试表明，启用它没有显著的性能提升，与原始工作中报告的结果一致；其次，这个设置隔离了动态预算分配机制，使我们能专门评估其相较于 Vertical-Slash 模式中使用的固定分配的影响。我们使用与 Vertical-Slash 实现中相同的任务依赖近似窗口（256 / 512 个 tokens）和关键 token 保留策略（前 4 个前缀，最近 64 个局部）。我们的消融实验（图 7 ）表明，将 `min_budget` 设置为 512 显著改善了性能，表明在预填充期间保持最低水平的连接性的重要性。我们通过基于注意力统计选择适当的 α 来实现目标压缩比，同时保持 `min_budget` 固定为 512。对于动态分配效果较低的高压缩比，我们设置 $\alpha = 0$ ，实际上恢复到 Vertical-Slash 的均匀分配。

我们通过在预填充阶段后压缩键值（KV）缓存，并在所有头之间应用统一的令牌预算，为之后的解码阶段实现 SnapKV。我们通过使用最近查询令牌的窗口（近似窗口）计算注意力分数来预测解码的令牌重要性。我们的消融实验显示出 256 个令牌的最佳近似窗口大小（图 8），且未观察到显著的任务依赖，与 Vertical-Slash 和 FlexPrefill 不同。我们使用尺寸为 21 的 1D 平均池化（根据图 9 选择）来平滑计算出的令牌重要性分数。我们始终保留最初的 4 个令牌和最近的 128 个令牌。我们通过设置 ‘token_capacity‘（每个头的令牌限制）来控制稀疏性，以匹配目标压缩率。

我们实现了 Ada-SnapKV，该方法通过为每个头部动态分配令牌预算扩展了 SnapKV。在我们实现的 Ada-SnapKV 和 SnapKV 之间的一个区别是，我们在评分计算中使用了最大聚合（而不是平均），跨查询位置和头部进行；这在经验上被证明对自适应分配更有效，但对均匀（SnapKV）分配没有影响。我们使用与 SnapKV 实现中相同的平滑核大小（21）和关键令牌保存策略（前 4 个前缀，最近的 128 个本地）。我们的消融实验（图 6 ）表明，为每个头部提供其容量的 20 % 的最低预算是最佳的。与 FlexPrefill 的敏感性相比，性能对最小预算的敏感性较小（在 10-50 % 范围内表现良好），但在接近 100 %（均匀分配）时性能急剧下降，这凸显了在解码期间动态分配的好处。我们通过设置 ‘token_capacity‘ 来控制稀疏性，与 SnapKV 相同。

A.1.3 探求

我们实现了 Quest (Tang et al., 2024) ，它在页面级别的解码阶段应用动态稀疏注意力。基于我们的消融实验（图 5 ），我们使用了 16 个标记的页面大小。我们通过页面的最小和最大键值来表示页面，以便能够与查询高效地计算相似性。在每个解码步骤中，我们根据查询-页面相似性分数选择最相关的页面，总是包括包含当前标记的页面。我们通过设置 ‘token_budget‘（每步选择的标记数）来控制稀疏性，以实现目标压缩比。

Table 6: 不同序列长度和压缩比的模式参数

Pattern	Parameter	Sequence Length	Values for Different Compression Ratios
Vertical & Slash	Verticals/Slashes	16384	164, 240, 315, 400, 448, 576, 768, 1024, 1536, 2304
		32768	290, 384, 448, 576, 704, 1024, 1536, 2304, 3584, 4608
		65536	400, 448, 544, 640, 960, 1280, 2304, 4096, 6144, 8192
		128000	480, 768, 1024, 1536, 2048, 3584, 5632, 10240, 13312, 18432
FlexPrefill	(α , min_budget)	16384	(0, 164), (0, 240), (0, 315), (0, 400), (0.55, 512), (0.71, 512), (0.88, 512)
		32768	(0, 290), (0, 384), (0.45, 512), (0.6, 512), (0.7, 512), (0.8, 512), (0.92, 512)
		65536	(0, 400), (0.45, 512), (0.55, 512), (0.7, 512), (0.77, 512), (0.85, 512), (0.94, 512)
Block Sparse	top_chunks	16384	26, 35, 53, 71, 108, 188, 300
		32768	52, 69, 105, 141, 216, 376, 600
		65536	104, 139, 210, 283, 432, 752, 1200
SnapKV/AdaSnapKV	token_capacity	16384	819, 1092, 1638, 2183, 3276, 4915, 6553, 8192, 9830, 11468
		32768	1638, 2185, 3276, 4367, 6553, 9830, 13107, 16384, 19660, 22937
		65536	3276, 4371, 6553, 8735, 13107, 19660, 26214, 32768, 39321, 45875
		128000	6400, 8544, 12800, 17056, 25600, 38400, 51200, 64000, 76800, 89600
Quest	token_budget	16384	816, 1088, 1632, 2176, 3280, 4912, 6560, 8192, 9824, 11472
		32768	1632, 2192, 3280, 4368, 6560, 9824, 13104, 16384, 19664, 22944
		65536	3280, 4368, 6560, 8736, 13104, 19664, 26208, 32768, 39328, 45872
		128000	6400, 8544, 12800, 17056, 25600, 38400, 51200, 64000, 76800, 89600

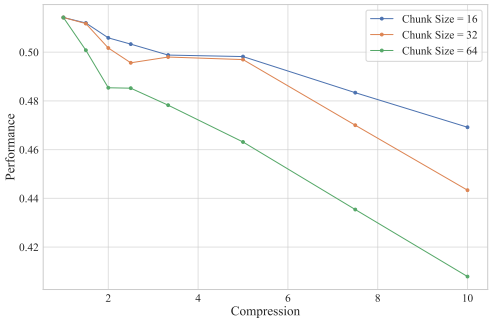


Figure 4: 块稀疏块大小。

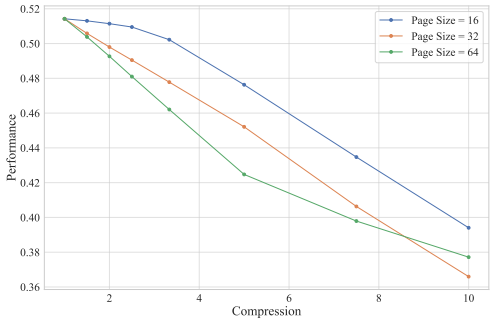


Figure 5: 问卷页面大小。

A.2 任务细节

本节提供了我们实验中使用的九个评估任务的详细信息。任务分为问答、来自 RULER (Hsieh et al., 2024) 的合成任务以及我们的故事任务。我们详细说明了关键的超参数、评估指标，并根据 Table 2 中定义的范围（低与高）和离散度（低与高）的轴线描述每个任务。范围是指所需信息的数量，而离散度表示在上下文中定位相关信息的难易程度。

我们使用来自 RULER 的 SQuAD (Rajpurkar et al., 2018) 和另外两个为最小数据污染而选择的 QA 数据集 (Li et al., 2024a)： QuALITY (Pang et al., 2022) 和 ToeflQA (Tseng et al., 2016) 。

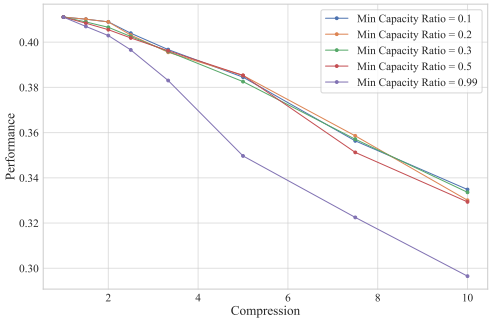


Figure 6: Ada-SnapKV 最小预算。

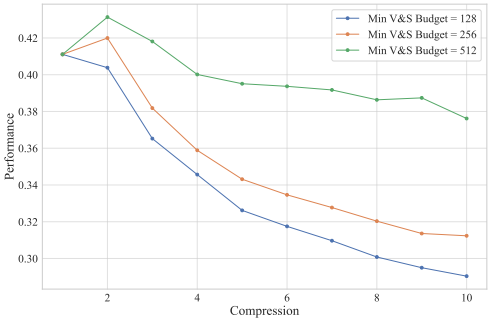


Figure 7: FlexPrefill 最小预算。

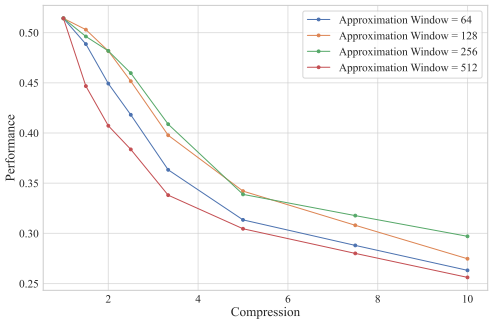


Figure 8: SnapKV/Ada-SnapKV 近似窗口。

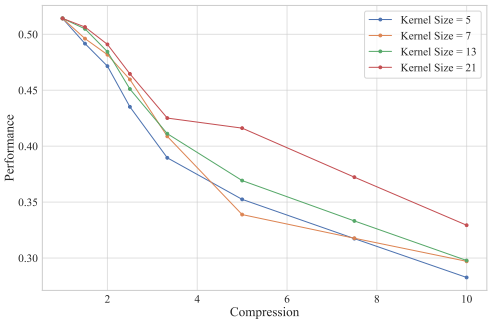


Figure 9: SnapKV/Ada-SnapKV 内核大小。

我们使用了来自 RULER 基准的三个合成任务。

这些任务使用程序生成的多章节叙述，随着序列长度的增长而扩展。每个章节都遵循一个包含旅行、对话和物品交易的模式。请参阅 Appendix F 以了解示例叙述。

- 故事检索：回答关于特定章节的 16 个事实性问题（例如，访问的地点，获得的物品），问题中提供了章节 ID。通过精确匹配准确率进行评估。需要访问特定章节，具有低分散性和低范围的特征。见 Appendix G.5 。
- 故事过滤：识别三个特定的章节，这些章节中没有发生任何物品购买。提示明确要求提供这三个章节的 ID，并构建叙述，以确保准确有三个章节符合此条件。使用 IoU 进行评估。需要检查所有章节，要求低分散性（信息是基于章节的）但高范围（必须检查所有章节）。我们发现，即便是对于评估过的最大模型，这个任务仍然是具有挑战性的。参见 Appendix G.6 。
- 故事多跳：给定一个目标物品，识别在其之前立即获得的物品，这需要在多个章节的交易历史中进行推理。在我们的设定中，每个章节都会获得一个物品；这将任务简化为定位目标物品被获得的章节，并从前一个章节中获取物品名称。我们发现，即使是对评估的最大模型而言，这种简化版本也是非常具有挑战性的，因此我们没有探讨更复杂的变体（例如，需要更长回溯的选择性物品获取）。使用精确匹配准确性进行评估。需要跟踪叙述中的历史，要求高分散性（相关交易可能相距甚远）和低范围（只有特定交易对才重要）。请参见 Appendix G.7 。

B 尺度律研究概述

规模定律研究旨在建模模型性能与各种因素之间的关系，例如模型大小、训练数据和推理预算。正如 Li et al. (2025a) 全面讨论的那样，这些研究通常涉及将性能指标与这些因素进行回归，使用系统实验中的数据。我们简要概述了该领域的关键方面，以便将我们的贡献置于更广泛的规模定律背景中。

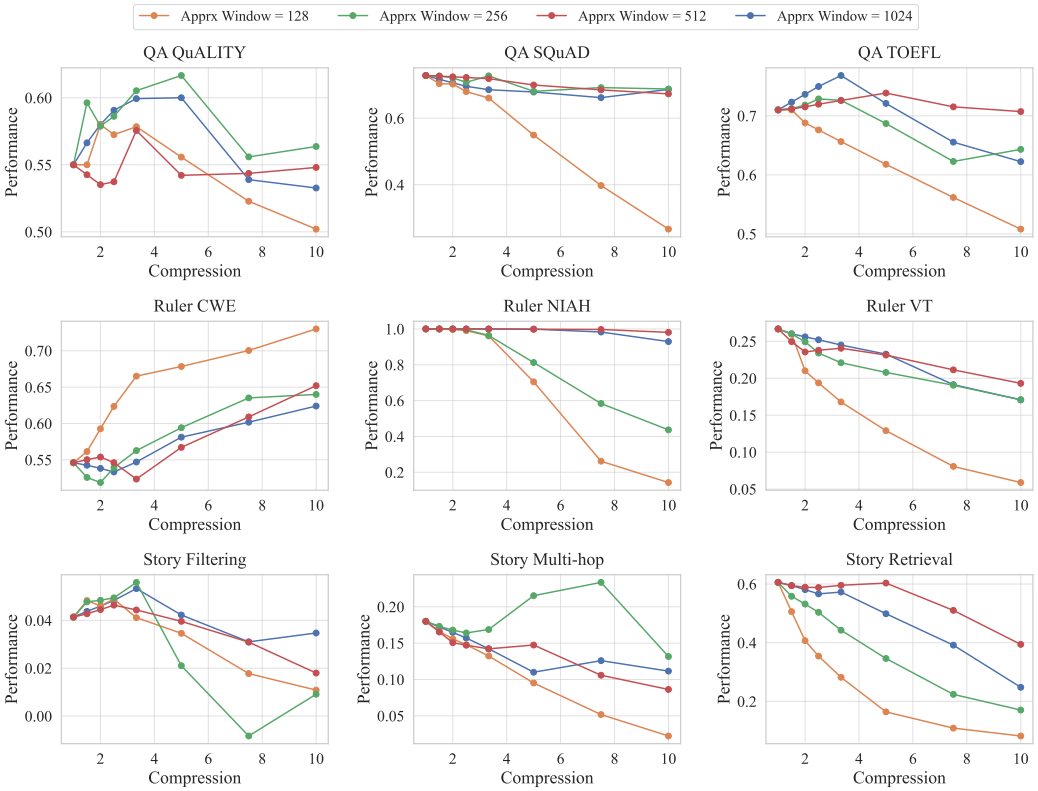


Figure 10: 每个任务的垂直斜杠逼近窗口消融。

性能指标 目标度量的选择取决于被控制的因素。对于研究模型规模或数据量变化的情况，语言模型指标如困惑度是常用的。对于监督模型或特定能力研究（例如，Yue et al. (2025) 中的 RAG 性能），通常使用任务准确性。在我们的工作中，我们使用长上下文基准（QA, RULER, Story）上的下游准确性来衡量性能，这直接评估了我们希望建模的能力。

大多数关于缩放规律的研究集中在训练计算上，这与模型大小和参数数量一起扩展。最近的工作已经扩展到研究模型稀疏性（模型专家稀疏性或激活稀疏性）、量化精度和推理预算。我们的工作在这个最后的类别中作出贡献，通过控制注意力稀疏性（通过压缩比），直接影响推理计算和内存需求，同时还有模型大小和序列长度。

实验方法 缩放法则数据可以通过控制实验 (Kaplan et al., 2020; Hoffmann et al., 2022) 或者分析不同家族的现有模型 (Choshen et al., 2024; Ruan et al., 2024) 来收集。前一种方法成本高但精确，而后一种方法利用公共数据，但可能受到未披露的训练程序差异的影响。我们在 Qwen 2.5 系列上进行了一系列稀疏注意力算法、稀疏级别（压缩比）、序列长度和任务的控制实验。

拟合方法 适合于拟合缩放规律的常见方法包括：

- 观察性能趋势以确定合适的函数形式（例如，幂律、指数）(Luo et al., 2025) 。我们采用对数线性形式，假设数值因子是幂律关系。
- 使用优化算法（通常为 L-BFGS）配合稳健的损失函数如 Huber 损失 (Abnar et al., 2025) 拟合参数。我们使用 L-BFGS 配合 Huber 损失。
- 通过细致的损失函数参数化 (Frantar et al., 2023) 或选择性数据过滤 (Li et al., 2025a) 来管理异常值。我们使用的 Huber 损失帮助减轻了异常值的影响。
- 在保留的例子上进行验证，特别是在参数范围 (Yue et al., 2025) 的极端值上。我们通过保留每个轴（模型大小、序列长度、压缩比）的最大值来进行验证。

我们的方法与这些既定的实践保持一致，具体如主文和附录 C 所述。

C 稀疏注意力缩放定律：公式与拟合细节

本节提供了我们在制定和拟合稀疏注意力缩放定律方面的方法的进一步细节，补充了主论文中的讨论（Section 4.4 ）。我们旨在根据模型大小、序列长度、压缩比和任务特性预测下游任务的准确性，着重于 Qwen 2.5 模型系列和表现最佳的稀疏模式（用于预填充的 Vertical-Slash，用于解码的 Quest）。

C.1 尺度定律公式

受到之前工作的启发，特别是 Yue et al. (2025)，我们制定了一种适合于建模长上下文任务推理性能的对数线性缩放法则公式：

我们将任务准确率 $a \in [0,1]$ 转化为其 logit $s = \sigma^{-1}(a) = \log(a) - \log(1-a)$，以创建一个无界的回归目标，适合线性建模。

模型大小 我们将参数数量 N 的影响建模为 $\alpha \log N$ 。基于一般的观察，较大的模型表现更佳，预计 α 是正的。

序列长度 我们包含一个项 $\beta \log L$，其中 L 是序列长度。由于较长的上下文通常带来更大的挑战，预计 β 会是负值。

压缩比 我们将压缩率 $C = \text{FLOPS}_{\text{dense}}/\text{FLOPS}_{\text{sparse}}$ 作为 $\gamma \log C$ 引入。由于较高的压缩（更多的稀疏性）通常会降低性能，预计 γ 会是负值。

我们包括一个任务特定的截距参数 δ_{task}，以解释任务难度和对其他因素敏感性方面的固有差异。

我们包括一个共享的截距 ϵ，表示基线性能水平。

正如主文中所述的完整缩放律公式是：

$$s = \alpha \log N + \beta \log L + \gamma \log C + \delta_{task} + \epsilon \tag{2}$$

C.2 拟合过程

我们使用 L-BFGS 算法优化缩放律参数（$\alpha, \beta, \gamma, \{\delta_{task}\}, \epsilon$ ）。为了提高对性能数据中潜在离群值的鲁棒性，我们使用带有 $\delta = 1$ 的 Huber 损失函数。正如主论文所述，我们进行了三次独立拟合运行以进行外推分析，每次都排除与某一个轴（模型大小、序列长度或压缩比）的最大值对应的数据点。我们也评估了均方误差损失，发现了相似的结果，并且在哪些任务和外推轴更难以准确建模方面表现出一致的模式。

C.3 缩放律准确性 (MAE)

正如主论文中所讨论的，我们使用 Spearman 的 ρ (表格 4) 和平均绝对误差（MAE）来评估我们拟合的缩放规律的外推能力，测试集是留置的。表 7 报告了不同外推轴（保留最大模型大小、序列长度或压缩比）之间各个任务的预测准确性（从 logit 空间转换回来）与真实准确性之间的 MAE。

较低的 MAE 表示更好的预测准确性。虽然整体拟合度很高（高 R^2，见表 3 ），但是 MAE 值突显了一些特定的任务-轴组合，在这些组合中外推更加具有挑战性。例如，在预填充阶段放弃最高压缩比的数据时，其在故事检索中的 MAE 相对较高（0.41），而在解码阶段，标尺 NIAH 的 MAE 相对较高（0.29）。然而，没有单一的任务或外推轴在预填充和解码两者中都普遍较难。一个有趣的观察是，在预填充阶段 MAE 较高的任务-轴对也倾向于在解码阶段表现出较高的 MAE（例如，序列长度外推中的故事检索），这表明某些任务对特定因素的固有敏感性在无论稀疏性是应用于预填充还是解码时都保持不变。

Table 7: 在沿不同轴线外推时（保留最大值），任务间预测和真实准确度的平均绝对误差（MAE）。越低越好。

		QA			Ruler			Story		
		QuALITY	SQuAD	TOEFL	CWE	NIAH	VT	Filtering	Multi-hop	Retrieval
Prefill	Model Size	0.16	0.11	0.06	0.11	0.02	0.22	0.24	0.08	0.07
	Sequence Length	0.12	0.08	0.10	0.23	0.16	0.06	0.04	0.06	0.20
	Compression Ratio	0.13	0.10	0.12	0.18	0.24	0.10	0.02	0.05	0.41
Decode	Model Size	0.19	0.15	0.08	0.07	0.04	0.29	0.19	0.19	0.02
	Sequence Length	0.16	0.12	0.17	0.14	0.19	0.08	0.04	0.07	0.24
	Compression Ratio	0.10	0.10	0.08	0.12	0.29	0.10	0.07	0.05	0.15

D FLOPS 分解

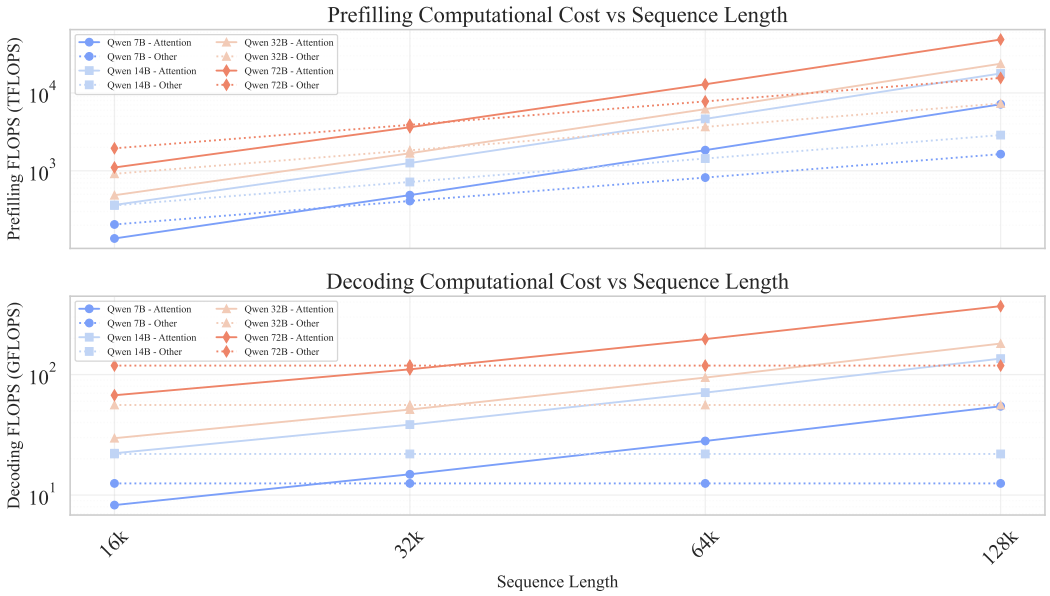


Figure 11: Qwen 模型（7B 到 72B）的计算成本在序列长度从 16K 到 128K 的范围内。顶部：预填充 FLOPS（TFLOPS）显示注意力（实线）和非注意力组件（虚线）。底部：解码 FLOPS（GFLOPS）有类似的分解。当线交叉时显示注意力 FLOPS 等于非注意力 FLOPS 的序列长度。

我们使用组件方法估算了来自 Qwen 2.5 系列模型的计算成本：

$$\text{FLOPS}_{\text{total}} = \text{FLOPS}_{\text{embedding}} + \text{FLOPS}_{\text{attention}} + \text{FLOPS}_{\text{mlp}} + \text{FLOPS}_{\text{logits}} \tag{3}$$

$$\text{FLOPS}_{\text{embedding}} = 2 \cdot L \cdot d \tag{4}$$

$$\text{FLOPS}_{\text{attention}} = N \cdot (2Ld^2 + 2L^2d \cdot \rho + 3hL^2 \cdot \rho + 2L^2d \cdot \rho) \tag{5}$$

$$\text{FLOPS}_{\text{mlp}} = N \cdot (4Ld \cdot d_{\text{mlp}} + 2L \cdot d_{\text{mlp}}) \tag{6}$$

$$\text{FLOPS}_{\text{logits}} = 2 \cdot L \cdot d \cdot |V| \tag{7}$$

其中，L 是序列长度，d 是隐藏维度，h 是头的数量，N 是层的数量，d_{mlp} 是 MLP 的中间维度，$|V|$ 是词汇大小，ρ 代表注意力密度（1-稀疏性）。对于解码，典型的生成长度（在我们的情况下平均为 200 个标记）比预填充长度短了几个数量级。因此，我们通过计算在关注整个长度为 L 的上下文时处理一个新标记的代价来估计解码的 FLOPS。

对于稀疏注意力方法，我们还考虑了重要性估计的计算成本——确定需要计算哪些注意力矩阵子集的过程。对于 Vertical-Slash 模式，这包括键和查询子集之间的点积、softmax 运算、对查询和对角线的聚合、top-k 选择以及索引构建。索引的成本大约为：

$$\text{FLOPS}_{\text{VS indexing}} = N \cdot h \cdot [2dLq + 3Lq + 2Lq + 2L\log_2(L) + \frac{L}{64}(k_v + k_s)] \tag{8}$$

其中，q 是用于重要性估计的最后几个查询的数量，k_v 和 k_s 是选择的垂直和斜杠模式的数量。

在解码过程中的 Quest，对于索引成本涉及到计算 KV 缓存中每个页面的重要性得分：

$$\text{FLOPS}_{\text{Quest indexing}} = N \cdot h \cdot [2d \cdot \frac{L}{p} + 3d \cdot \frac{L}{p} + \frac{L}{p}\log_2(\frac{L}{p})] \tag{9}$$

其中 p 是页面大小（在我们的案例中为 16）。这些索引成本包括在我们的 isoFLOPS 分析中，以确保密集注意力方法和稀疏注意力方法之间的公平比较。

图 11 展示了预填充和解码操作所需的 FLOPS，并揭示了在 transformer 模型中的两种不同计算机制：一是在较短序列（低于 16K）时的线性主导机制，其中非注意力组件（如嵌入层、MLP 和输出层）决定了计算成本；二是在较长序列（超过 16K-48K，具体取决于模型大小）时的二次方主导机制，其中注意力计算及其 $O(L^2)$ 复杂性成为推动总计算需求的主要因素。

在以二次函数为主的范畴中，一个重要的见解是：在足够长的序列长度下，具有稀疏注意力的较大模型所需的 FLOPS 可能少于较小的密集模型。这种效率交叉点之所以出现，是因为稀疏性大幅度减少了注意力的开销，而这种开销的增长速度快于线性部分。随着序列长度的增加，这种效应愈发明显，使得稀疏大模型在计算上比密集小模型愈加具有优势。

E 提示模板

Input format:

```
You are provided with a task introduction, context, and a question.

{task_intro}

Below is your question. I will state it both before and after the context.

<question>
{question}
</question>

<context>
{context}
</context>

<question_repeated>
{question}
</question_repeated>

Instructions:
1. First, provide a brief explanation of your reasoning process. Explain how you identified
   the relevant information from the context and how you determined your answer.
2. Then, provide your final answer following this exact format:
<answer>
{answer_format}
</answer>

Your response must follow this structure exactly:
<explanation>
Your explanation here...
</explanation>
<answer>
Your answer here...
</answer>

Important:
{extra_instructions}
- Keep your explanations clear, coherent, concise, and to the point.
- Do not include any additional text, explanations, or reasoning in the answer section.
```

F 示例故事叙述

Chapter 1:
Beneath gentle breezes, Arion ventured into Athens, curious about its secrets. Long journeys had led Arion to Athens, a step closer to understanding. Soon enough, a tense negotiation seized everyone's attention. Cleo appeared as if expecting Arion, engaging them without delay. Carefully, they navigated the topic of old feuds, wary of awakening dormant animosities that still simmered. In a calm moment, they compared notes on the traders who passed through Athens, each leaving their subtle mark. In hushed tones, they spoke of local customs and distant rumors, sharing hints of hidden pathways. Following subtle bargaining with Cleo, Arion claimed ownership of lavish crystal lamp. With a light gesture, Arion acknowledged Cleo once more before departing. Nothing would be the same as Arion left Athens, thoughts turning inward. In quiet corners, ambitions simmered, waiting for a spark.

Chapter 2:
At dawn, Arion reached the gates of Hippo Regius, where merchants and travelers converged. This place might hold a clue Arion had long sought. Hardly had Arion arrived before a violent storm stirred uneasy whispers. Thanos approached Arion, eyes bright with opportunity. They lingered over tales of old alliances and forgotten disputes, weaving past into present. They debated the meaning of recent events, each seeking patterns in the chaos. Their reflections turned to the interplay of supply and demand, seeing how fortunes might turn in an instant. After reaching terms with Thanos, Arion took possession of ceremonial gold seal. Arion turned from Thanos, ready to move on. In parting, Arion acknowledged that the journey still had far to run. Hidden corners of the city promised knowledge or peril.

Chapter 3:
The threshold of Emerita Augusta welcomed Arion, who felt the weight of untold stories. Arion came here hoping to learn something new, or perhaps gain an advantage. Within hours, a violent storm disrupted the familiar routines. There, Arion encountered Niko, who seemed eager to exchange words or goods. Their words lingered on rumors of distant lands, where fortunes or ruin awaited bold seekers. They debated the meaning of recent events, each seeking patterns in the chaos. Their dialogue danced around subtle clues, each suggestion hinting at treasures undiscovered. The transaction concluded with Arion acquiring delicate porcelain sword from Niko. With a light gesture, Arion acknowledged Niko once more before departing. Eventually, Arion moved on, carrying new impressions forward. The distant hum of voices hinted at unseen deals.

Chapter 4:
Under fading daylight, Arion set foot in Berenice, eager to learn what it offered. A quiet determination brought Arion to Berenice, ever searching for meaning. a sudden market crash cast its shadow over Berenice, changing plans and minds. Roxana approached Arion, eyes bright with opportunity. Together, they reflected on the nature of trust and deceit, aware that fate often twists. They compared accounts of strange visitors bearing knowledge or confusion, each arrival a new riddle in Berenice. A short exchange revealed uncharted corners of Berenice, where knowledge or secrets might dwell. mystic bronze lamp changed hands as Arion completed the purchase from Roxana. Arion handed over lavish crystal lamp to Roxana as the deal closed. With a light gesture, Arion acknowledged Roxana once more before departing. As Arion prepared to depart, the path ahead remained uncertain but compelling. Somewhere, a whisper promised answers for those who dared.

Chapter 5:
Under fading daylight, Arion set foot in Syracuse, eager to learn what it offered. In pursuit of truth, Arion looked to Syracuse for subtle revelations. Not long after arriving, an opulent banquet shook the local order. Phaedra appeared as if expecting Arion, engaging them without delay. Their words traced over delicate negotiations that had once sealed lasting truces in Syracuse. Carefully, they navigated the topic of old feuds, wary of awakening dormant animosities that still simmered. They delved into the subtle art of earning trust in a place where trust was scarce and hard-won. With measured consideration, Arion purchased engraved emerald goblet from Phaedra, examining it closely. In quiet understanding, Arion left Phaedra, their paths diverging. In parting, Arion acknowledged that the journey still had far to run. A subtle tension lingered, as though fate held its breath.

G 示例任务输入

G.1 问答系统 (QA)

Input format:

```
I will provide you with multiple documents and ask you a question about one specific document.

Below is your question. I will state it both before and after the context.

<question>
Question about document 39:
Who works to get workers higher compensation?
</question>

<context>
Document 1:
[...text omitted...]

Document 39:
Jobs with high demand and low supply pay more. Professional and labor organizations
can raise wages by limiting worker supply and using collective bargaining or political influence.

Document 47:
[...text omitted...]
</context>

<question_repeated>
Question about document 39:
Who works to get workers higher compensation?
</question_repeated>

Instructions:
1. Provide a brief explanation of your reasoning process.
2. Then, give your final answer in this format:
<answer>
Your answer here...
</answer>

Your response must follow this structure:
<explanation>
Your explanation here...
</explanation>
<answer>
Your answer here...
</answer>

Important:
- Do not use complete sentences in the answer.
- For dates: Include ONLY the COMPLETE date if specifically asked.
- For locations: Use the shortest unambiguous form (e.g., 'New York' not 'New York City').
- For comparisons: State ONLY the answer that matches the criteria
- Keep your explanations clear, coherent, concise, and to the point.
- Do not include any additional text, explanations, or reasoning in the answer section.
```

Example answer:

```
<explanation>
I found the relevant sentence in document 39, which states that professional and labor
organizations help increase wages using bargaining and political means.
</explanation>
<answer>
Professional and labor organizations
</answer>
```

G.2 RULER - 大海捞针 (NIAH)

Input format:

```
I will provide you with a document containing multiple key-value pairs.
Your task is to extract specific values associated with given keys.

Below are your questions. I will state them both before and after the context.

<questions>
Extract the values for the following keys:
key-A, key-B, key-C, key-D
</questions>

<context>
The value for key-A is: value-A.
The value for key-X is: value-X.
The value for key-B is: value-B.
The value for key-Y is: value-Y.
The value for key-C is: value-C.
The value for key-Z is: value-Z.
The value for key-D is: value-D.
</context>

<questions_repeated>
Extract the values for the following keys:
key-A, key-B, key-C, key-D
</questions_repeated>

Instructions:
1. First, provide a brief explanation of your reasoning process. Explain how you identified
   the relevant information from the context and how you determined your answer.
2. Then, provide your final answer following this exact format:
<answer>
1. The answer for <key1> is <value1>.
2. The answer for <key2> is <value2>.
etc.
</answer>

Your response must follow this structure exactly:
<explanation>
Your explanation here...
</explanation>
<answer>
Your answer here...
</answer>

Important:
- Provide answers in the exact order of the requested keys
- Each answer must follow the format: "<number>. The answer for <key> is <value>."
- Ensure exact key matches - do not modify or paraphrase the keys
- Values must match exactly as they appear in the document
- Keep your explanations clear, coherent, concise, and to the point.
- Do not include any additional text, explanations, or reasoning in the answer section.
```

Example answer:

```
<explanation>
I scanned the context for exact matches of the requested keys. For each key, I extracted
the value as stated directly after the pattern "The value for key-X is: ...".
</explanation>
<answer>
1. The answer for key-A is value-A.
2. The answer for key-B is value-B.
3. The answer for key-C is value-C.
4. The answer for key-D is value-D.
</answer>
```

G.3 RULER - 常用词提取 (CWE)

Input format:

```
You will be given a numbered list of words. Your task is to identify the most
frequently occurring words. You should solve this task by carefully reading
and analyzing the word list. Do not attempt to write code or use programming
tools to count frequencies. This is a test of your ability to track word
frequencies directly.

Below is your question. I will state it both before and after the context.

<question>
The list contains exactly 10 words that appear 30 times each.
All other words appear 3 times each.
Your task is to identify the 10 words that appear 30 times each.
</question>

<context>
1. alpha
2. beta
3. gamma
4. delta
5. alpha
6. epsilon
...
[...list continues with randomized repeated words...]
...
N. gamma
</context>

<question_repeated>
The list contains exactly 10 words that appear 30 times each.
All other words appear 3 times each.
Your task is to identify the 10 words that appear 30 times each.
</question_repeated>

Instructions:
1. First, provide a brief explanation of your reasoning process.
   Explain how you identified the relevant information from the context
   and how you determined your answer.
2. Then, provide your final answer following this exact format:
<answer>
1. word_one
2. word_two
...
10. word_ten
</answer>

Your response must follow this structure exactly:
<explanation>
Your explanation here...
</explanation>
<answer>
Your answer here...
</answer>

Important:
- List exactly 10 words, one per line, numbered from 1 to 10.
- Keep your explanations clear, coherent, concise, and to the point.
- Do not include any additional text, explanations, or reasoning in the answer section.
```

Example answer:

```
<explanation>
I scanned the word list and tracked the frequency of each word.
The following 10 words appeared 30 times each, which I confirmed by careful counting.
</explanation>
<answer>
1. diligent
2. ash
3. pour
4. chateau
5. marble
6. laparoscope
7. grub
8. vinyl
9. mobility
10. kettledrum
</answer>
```

G.4 RULER - 变量跟踪 (VT)

Input format:

```
I will provide you with a text containing variable assignments. The text contains two types of assignments:
1. Numeric assignments that set a variable to a number (e.g., "VAR ABC = 12345")
2. Copy assignments that set a variable equal to another variable (e.g., "VAR XYZ = VAR ABC")
Variables are sequences of uppercase letters. The assignments can appear in any order in the text.

Below is your question. I will state it both before and after the context.

<question>
Which variables resolve to the value 41015? A variable resolves to 41015 if it is either directly assigned
41015, or assigned to another variable that resolves to 41015.
</question>

<context>
VAR A = VAR B
VAR B = 41015
VAR C = VAR D
VAR D = VAR B
VAR E = 12345
VAR F = VAR G
VAR G = VAR H
VAR H = VAR B
</context>

<question_repeated>
Which variables resolve to the value 41015? A variable resolves to 41015 if it is either directly assigned
41015, or assigned to another variable that resolves to 41015.
</question_repeated>

Instructions:
1. First, provide a brief explanation of your reasoning process. Explain how you identified
   the relevant information from the context and how you determined your answer.
2. Then, provide your final answer following this exact format:
<answer>
VARIABLE_ONE VARIABLE_TWO etc.
</answer>

Your response must follow this structure exactly:
<explanation>
Your explanation here...
</explanation>
<answer>
Your answer here...
</answer>

Important:
- List ONLY the variable names that resolve to the target value.
- Variables can be listed in any order.
- Do not include "VAR" prefix in your answer. Do not include punctuation.
- Keep your explanations clear, coherent, concise, and to the point.
- Do not include any additional text, explanations, or reasoning in the answer section.
```

Example answer:

```
<explanation>
I traced each variable assignment to see if it leads to the value 41015. B is directly assigned 41015.
A, D, and H point to B. C and G point to D and H, respectively. So A B C D G H resolve to 41015.
</explanation>
<answer>
A B C D G H
</answer>
```

G.5 故事检索

Input format:

```
You are given a narrative composed of multiple chapters. Throughout these chapters, the
protagonist travels between different locations, meets various characters, and engages
in trading activities. All items mentioned in the narrative are unique, and their
ownership can change through trades. Your task is to carefully read the narrative and
answer the questions based on the provided information.

Below are your questions. I will state them both before and after the context.

<questions>
1. In Chapter 3, which character did the protagonist interact with?
2. In Chapter 5, which specific item was acquired by the protagonist?
3. In Chapter 7, which specific location did the protagonist visit?
</questions>

<context>
Chapter 1:
[...text omitted...]

Chapter 3:
Arion entered Babylon and met Thanos. After exchanging stories, Arion acquired a silver idol.

Chapter 5:
In Berenice Troglodytica, Arion encountered Xanthe and traded for a golden vase.

Chapter 7:
Delphi welcomed Arion with quiet mystery. A meeting with Vitalis ended with a jade idol.
</context>

<questions_repeated>
1. In Chapter 3, which character did the protagonist interact with?
2. In Chapter 5, which specific item was acquired by the protagonist?
3. In Chapter 7, which specific location did the protagonist visit?
</questions_repeated>

Instructions:
1. First, provide a brief explanation of your reasoning process. Explain how you identified
   the relevant information from the context and how you determined your answer.
2. Then, provide your final answer following this exact format:
<answer>
1. ANSWER_ONE
2. ANSWER_TWO
etc.
</answer>

Your response must follow this structure exactly:
<explanation>
Your explanation here...
</explanation>
<answer>
Your answer here...
</answer>

Important:
- For answers, use one line per answer with the number prefix
- Do not include articles like 'the' or 'a' in answers
- Answers should be specific names/items/locations mentioned in the text
- Keep your explanations clear, coherent, concise, and to the point.
- Do not include any additional text, explanations, or reasoning in the answer section.
```

Example answer:

```
<explanation>
I located Chapter 3 in the context and identified Thanos as the mentioned character.
In Chapter 5, Arion acquired a golden vase from Xanthe.
Chapter 7 stated that Arion visited Delphi, so I used that as the answer.
</explanation>
<answer>
1. Thanos
2. Golden Vase
3. Delphi
</answer>
```

G.6 故事筛选

Input format:

```
You are given a narrative composed of multiple chapters. Throughout these chapters, the
protagonist travels between different locations, meets various characters, and engages
in trading activities. All items mentioned in the narrative are unique, and their
ownership can change through trades. Your task is to carefully read the narrative and
answer the questions based on the provided information.

Below is your question. I will state it both before and after the context.

<question>
Identify all chapters where the protagonist did not buy any item.
Note: There are exactly 2 chapters without any purchases.
</question>

<context>
Chapter 1:
[... Arion visits Athens and purchases a crystal lamp ...]

Chapter 2:
[... Arion travels to Hippo Regius and buys a gold seal ...]

Chapter 3:
[... Arion enters Babylon and engages in an ongoing event but do not buy anything ...]

Chapter 4:
[... Arion arrives in Pergamon and has conversations, but no purchases are mentioned ...]

Chapter 5:
[... Arion goes to Delphi and buys a jade idol ...]
</context>

<question_repeated>
Identify all chapters where the protagonist did not buy any item.
Note: There are exactly 2 chapters without any purchases.
</question_repeated>

Instructions:
1. First, provide a brief explanation of your reasoning process. Explain how you identified
   the relevant information from the context and how you determined your answer.
2. Then, provide your final answer following this exact format:
<answer>
chapter_id_1, chapter_id_2, ...
</answer>

Your response must follow this structure exactly:
<explanation>
Your explanation here...
</explanation>
<answer>
Your answer here...
</answer>

Important:
- In the answer section, provide only the chapter IDs separated by commas.
- Keep your explanations clear, coherent, concise, and to the point.
- Do not include any additional text, explanations, or reasoning in the answer section.
```

Example answer:

```
<explanation>
I scanned each chapter to check whether a purchase by the protagonist was explicitly
described. In Chapter 3 and 4, no item acquisition are mentioned. Other chapters include
phrases like "Arion purchased" or "Arion acquired", indicating a transaction.
</explanation>
<answer>
3, 4
</answer>
```

G.7 故事多跳

Input format:

```
You are given a narrative composed of multiple chapters. Throughout these chapters,
the protagonist travels between different locations, meets various characters,
and engages in trading activities. All items mentioned in the narrative are unique,
and their ownership can change through trades. Your task is to carefully read the
narrative and answer the questions based on the provided information.

Below is your question. I will state it both before and after the context.

<question>
What was the last item that the protagonist acquired before acquiring timeworn amber sword?
</question>

<context>
Chapter 1:
[... narrative text omitted for brevity ...]

Chapter 17:
The transaction concluded with Arion acquiring pristine bronze seal from Damon.

Chapter 18:
After reaching terms with Marcus, Arion took possession of timeworn amber sword.
</context>

<question_repeated>
What was the last item that the protagonist acquired before acquiring timeworn amber sword?
</question_repeated>

Instructions:
1. First, provide a brief explanation of your reasoning process. Explain how you identified
   the relevant information from the context and how you determined your answer.
2. Then, provide your final answer following this exact format:
<answer>
ITEM_NAME
</answer>

Your response must follow this structure exactly:
<explanation>
Your explanation here...
</explanation>
<answer>
Your answer here...
</answer>

Important:
- Provide only the item name in the answer section.
- Do not include articles like 'the' or 'a' in your answer.
- The item name must be exactly as mentioned in the text.
- Keep your explanations clear, coherent, concise, and to the point.
- Do not include any additional text, explanations, or reasoning in the answer section.
```

Example answer:

```
<explanation>
I located the chapter where the protagonist acquired the timeworn amber sword.
Then, I scanned earlier chapters to find the most recent prior acquisition,
which occurred in Chapter 17 with the item pristine bronze seal.
</explanation>
<answer>
pristine bronze seal
</answer>
```